

RL-TR-97-165
Final Technical Report
October 1997



SERVICE CHARACTERISTICS BASED HIGH SPEED MULTIMEDIA TRANSPORT PROTOCOL

Syracuse University

C Y Roger Chen

DTIC QUALITY INSPECTED &

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

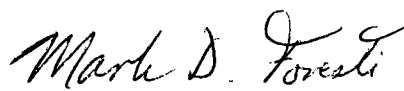
19980317 055

Rome Laboratory
Air Force Materiel Command
Rome, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-97-165 has been reviewed and is approved for publication.

APPROVED:



MARK D. FORESTI
Project Engineer

FOR THE DIRECTOR:



SAMUEL A. DINITTO, JR., Director
Command, Control & Communications Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL/C3, 52 Brooks Road, Rome, NY 13441-4505. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

| REPORT DOCUMENTATION PAGE | | | Form Approved OMB No. 0704-0188 | |
|---|---|--|--------------------------------------|--|
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503. | | | | |
| 1. AGENCY USE ONLY (Leave blank) | 2. REPORT DATE October 1997 | 3. REPORT TYPE AND DATES COVERED Final: Sep 94 - Oct 96 | | |
| 4. TITLE AND SUBTITLE SERVICE CHARACTERISTICS BASED HIGH SPEED MULTIMEDIA TRANSPORT PROTOCOL | | 5. FUNDING NUMBERS C - F30602-94-1-0007 PE - 62702F PR - 4600 TA - AO WU - A2 | | |
| 6. AUTHOR(S) Dr. C.Y. Roger Chen | | 8. PERFORMING ORGANIZATION REPORT NUMBER | | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Syracuse University Office of Sponsored Programs 113 Bowne Hall Syracuse, NY 13244-1200 | | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER RL-TR-97-165 | | |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Rome Laboratory/C3AB 525 Brooks Road Rome, NY 13441-4505 | | 11. SUPPLEMENTARY NOTES Rome Laboratory Project Engineer: Mark D. Foresti/C3AB/(315) 330-2233 | | |
| 12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release: Distribution Unlimited | | 12b. DISTRIBUTION CODE | | |
| 13. ABSTRACT (Maximum 200 words) This effort consists of three phases. Phase I will focus on conceptual design, modeling, and investigation of quality of service requirements for application environments. Phase II is centered on the actual development and system implementation of the protocol in a UNIX network programming environment. The final phase consists of the development and implementation of applications which will exploit the use of the new multimedia protocol and a demonstration of the system environment. This multimedia protocol will automatically extract service requirements from multimedia applications by selecting certain sets of parameters from user applications to optimally drive the lower level physical network. The development of protocols within such a system would assist in solving problems associated with continuous medial applications, such as video and information on demand. | | | | |
| 14. SUBJECT TERMS MPEG I, MPEG II, Multimedia, Transport Protocols, ATM, MPEG Video | | 15. NUMBER OF PAGES 134 | | |
| | | 16. PRICE CODE | | |
| 17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED | 18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED | 19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED | 20. LIMITATION OF ABSTRACT UL | |

Table of Contents

| | |
|--|-----------|
| Chapter 1. Introduction..... | 1 |
| 1.1 Characterization of Multimedia Applications..... | 3 |
| 1.1.1 Traffic Characteristics..... | 3 |
| 1.1.2 Communication Requirements..... | 4 |
| 1.2 MPEG Video Standard | 7 |
| 1.3 Contributions | 10 |
| Chapter 2. Lossless Smoothing Algorithms for VBR Traffic | 14 |
| 2.1 Introduction..... | 14 |
| 2.2 System Model | 17 |
| 2.2.1 Delay Constraints..... | 20 |
| 2.2.2 Buffer Constraints..... | 24 |
| 2.3 Smoothing Algorithm | 27 |
| 2.3.1 Shortest Path | 27 |
| 2.3.2 Causal Algorithm Design and Specification..... | 30 |
| 2.4 Experiments | 35 |
| 2.4.1 Performance of Original Causal Algorithm..... | 36 |
| 2.4.2 Improvement of Shortest Path Algorithm..... | 45 |
| 2.4.3 Comparison of Smoothing Algorithms..... | 46 |
| 2.4.4 Buffer vs Delay Constraint | 53 |
| 2.5 Conclusion | 57 |
| Chapter 3. Effects of Smoothing on End-to-end Deterministic Guarantees for VBR Traffic..... | 58 |
| 3.1 Introduction..... | 58 |
| 3.2 The Deterministic D-BIND Model | 62 |
| 3.3 Connection Admission Control | 65 |
| 3.4 Effect of Smoothing on the Deterministic Service | 66 |
| 3.4.1 The Single Hop Case | 68 |
| 3.4.2 The Multi-Hop Case | 70 |
| 3.5 Experimental Results | 72 |
| 3.5.1 Ideal Smoothing (Stored Video) | 73 |
| 3.5.2 Real-time Traffic..... | 78 |
| 3.6 Conclusion | 82 |
| Chapter 4. Aggregate Smoothing: Integration of Traffic Shaping and Multiplexing | 84 |
| 4.1 Introduction..... | 84 |
| 4.2 Specification of Aggregate Smoothing..... | 89 |

| | |
|---|------------|
| 4.3 Evaluation of Aggregate Smoothing..... | 96 |
| 4.4 Effect of Aggregate Smoothing on End-to-end Deterministic Guarantees..... | 103 |
| 4.5 Conclusion | 106 |
| Chapter 5. Conclusion and Future Work..... | 108 |
| 5.1 Overview of Presented Work..... | 109 |
| 5.2 Contributions | 111 |
| 5.3 Future Work..... | 112 |
| Bibliography..... | 114 |

List of Figures

| | |
|--|-----|
| Figure 1.1: An MPEG video sequence. | 9 |
| Figure 2.1: System model for smoothing algorithm. | 17 |
| Figure 2.2: Illustration of the shortest path algorithm. In (a) two chains are preserved and in (b) a new cusp is found. | 29 |
| Figure 2.3: Specification of the causal smoothing algorithm. | 31 |
| Figure 2.4: Specification of the shortest-path algorithm. | 34 |
| Figure 2.5: Two MPEG video sequences: Formula 1 and Star Wars. | 38 |
| Figure 2.6: Rate as a function of time for original bit stream and three delay bounds. | 39 |
| Figure 2.7: Delays of pictures in Formula 1 video sequence. | 40 |
| Figure 2.8: Performance of causal algorithm as a function of delay bound. | 42 |
| Figure 2.9: Performance of causal algorithm as a function of parameter H. | 43 |
| Figure 2.10: Performance of causal algorithm as a function of parameter K. | 44 |
| Figure 3.1: D-BIND rate-interval pairs for a segment of Goldeneye Movie. | 63 |
| Figure 3.2: D-BIND constraint function for Goldeneye sequence. | 64 |
| Figure 3.3: Deterministic Admission Control. | 65 |
| Figure 3.4: Effect of smoothing on network queuing delay bound. | 69 |
| Figure 3.5: Network topology used in the experiments. | 72 |
| Figure 3.6: Average utilization of the multiplexer for various smoothing delays. | 73 |
| Figure 3.7: Average utilization as a function of total end-to-end delay bound (a) for the single hop and (b) for three hops. | 74 |
| Figure 3.8: Effect of the number of hops on the savings in total delay bound. | 75 |
| Figure 3.9: Effect of network load. | 76 |
| Figure 3.10: Effect of smoothing delay for one-hop network. | 77 |
| Figure 3.11: Effect of smoothing delay for three-hops network. | 78 |
| Figure 3.12: Effect of smoothing delay on total end-to-end delay bound. | 78 |
| Figure 3.13: Utilization vs queuing delay for real-time smoothing. | 79 |
| Figure 3.14: Smoothing delay vs. Savings in the end-to-end delay bound for real-time and ideal smoothing algorithms. | 80 |
| Figure 3.15: Effect of smoothing delay on the end-to-end delay. | 80 |
| Figure 3.16: Minimum number of congested hops to obtain positive savings. | 81 |
| Figure 4.1: Applications of aggregate smoothing: (a) video broadcasting (b) Video-on- Demand (c) Video-telephony service. | 86 |
| Figure 4.2: System model for aggregate smoothing. | 90 |
| Figure 4.3: (a) Aggregate bit rate is out of bounds. (b) Closest rate vector is out of bounds. | 94 |
| Figure 4.4: (a) Rate function of total rate at the multiplexer output. (b) Rate function of a single stream. | 95 |
| Figure 4.5: Performance of the aggregate smoothing algorithm for stored video. | 100 |
| Figure 4.6: Performance of the aggregate smoothing algorithm for real-time video. | 101 |
| Figure 4.7: Effect of smoothing delay on the performance of aggregate smoothing. | 102 |

| | |
|--|-----|
| Figure 4.8: Scenario for trace-driven simulation of video aggregation scheme. | 104 |
| Figure 4.9: Effect of network utilization on (a) the savings in end-to-end delay bound (b) the end-to-end delay. | 105 |
| Figure 4.10: Effect of smoothing delay on the savings in end-to-end delay. | 106 |

Chapter 1

Introduction

Future high-speed broadband networks based on the Asynchronous Transfer Mode (ATM) technology will provide a large variety of services that cater to the needs of distributed multimedia applications [1, 2]. The term *distributed multimedia* has been used to describe the emerging scenario in which a single integrated network will carry a wide variety of media such as audio, video, image or plain data associated with traffic classes. Not only a broad range of traffic classes will be carried, but also a guaranteed *quality-of-service* (QoS) will be provided to some of these traffic classes. The issue of providing such QoS guarantees while taking advantage of the resource gains offered by a statistically multiplexed transport mechanism still remains as a challenging task for network architects. This task is further complicated by the fact that traffic generated by a multimedia application may not fall into a specific type of traffic class supported by the network due to the ever-growing number of new multimedia applications with diverse media type and QoS requirements. Therefore, traffic submitted to the network must be shaped in order to

receive maximum performance in terms of network QoS. For users¹ (network clients), traffic shaping allows for better utilization of available transport services and in general, it costs less because of the possible reduction in bandwidth (or allocated network resource) requirements. On the other hand, more connections can be admitted by the network for a given QoS due to higher network utilization. However, QoS requirements for applications are typically *end-to-end* requirements, which impose corresponding performance requirements on both the *network* and the *end-systems* (hosts). Thus, applications can impose a set of constraints on the traffic shaping function. In this report, such traffic shaping algorithms for a given set of constraints are introduced and the effect of traffic shaping on bandwidth allocation and network performance guarantee is investigated. In the next section, traffic characteristics of multimedia applications and their corresponding communication requirements are summarized. Then a short description of MPEG video standard is provided, since the bit streams used for the experiments in this report are based on MPEG encoded video sequences. A summary of ATM network services is introduced along with new services proposed in the literature but not yet standardized. Next, a discussion on how traffic shaping allows applications to efficiently utilize the network transport services is presented and the motivation of the research pursued in this report is explained. Finally, the chapter concludes by summarizing the main contributions of this report.

¹ The terms application, user and client will be used interchangeably through the rest of the report.

1.1 Characterization of Multimedia Applications

The various forms of multimedia data can be categorized as static and continuous [3]. While examples of static media include plain data, raw ASCII, numerical data, image and vector graphics, continuous media types include compressed or uncompressed audio and video, animation and interactive data all of which imply a temporal dimension. Multimedia applications can be characterized by their traffic characteristics and the corresponding communications requirements. An application's traffic characteristics can be described as one or more sequence of packets, of arbitrary length, generated at a certain time and destined for one or more locations. Each packet has an associated set of communications requirements. Traffic characteristics, together with the corresponding communications requirements, determine the network resources (bandwidth and buffer) needed to support this application.

1.1.1 Traffic Characteristics

The traffic characteristics of an application can be formally specified by its traffic generation process which is basically a sequence of packets at arbitrary instants, each packet having an arbitrary length. If packet generation occurs at regular time intervals, it is a periodic traffic pattern. If these packet lengths are fixed in size, it is a constant bit rate (CBR) traffic, otherwise it is a variable bit rate (VBR) traffic. For example, uncompressed audio and video streams are typically CBR traffic. On the other hand, compressed video is VBR in nature since the amount of compressed information varies according to

the content and instantaneous scene changes. This report focuses on VBR traffic as defined above.

1.1.2 Communication Requirements

Multimedia communication requirements have been extensively covered in [4, 5]. In the following, QoS parameters relevant to the underlying network services are summarized.

Bandwidth. The required bandwidth depends on whether data is compressed or not and if compressed, the encoding scheme especially for video and audio applications. Today's systems handle video data almost exclusively in compressed form in order to reduce transmission bandwidth and storage requirements. For a user-defined QoS level, it is possible to use one of the compression standards developed exclusively for audio and video. Some audio and video compression standards and their respective bandwidth requirements for particular applications are given in Table 1.1. It should be noted that the presented values are for average bandwidth since bandwidth can have peak and average values for VBR traffic. Three video compression standards have been widely accepted: International Standards Organization (ISO) Moving Pictures Expert Group (MPEG) [6], Intel's Digital Video Interactive (DVI), and International Telecommunications Union H.261 [7]. Practical experience with DVI and MPEG-1 suggests a total of 1.4 Mbps for audio and video, as it provides good video quality and accommodates commercial audiovisual equipment. The second phase of MPEG-1 known as MPEG-2 aims to address ap-

Table 1.1: Some audio and video compression standards and their respective bandwidth requirements for a given application.

| Standard | Bandwidth Requirement | Applications |
|---|--------------------------|--|
| ADPCM (CCITT G.723) | 24 Kbps | internet packetized voice communications |
| μ -law compressed PCM (CCITT G.711) | 64 Kbps | ISDN Digital Telephony Service |
| MPEG-1 Audio | 256 Kbps | 48-kHz-sampled stereo CD-quality audio |
| H.120 | 1544 Kbps or 2 Mbps | videoconferencing |
| H.261 | from 64 Kbps to 2 Mbps | videoconferencing, video-phony |
| MPEG-1 | ≤ 1.86 Mbps | CD-ROM, desktop video |
| MPEG-2 (Low) | 4 Mbps | CIF, VHS-quality |
| MPEG-2 (Main) | 15 Mbps | CCIR-601, studio TV |
| MPEG-2 (High 1440) | 60 Mbps | 4xCCIR-601, HDTV |
| MPEG-2 (High) | 80 Mbps | production SMPTE 240M standard |
| MPEG-4 | from 4.8 Kbps to 64 Kbps | dial-up video, videophone over phone lines |

plications at broadcast TV sample rates for higher quality video coded at around 4 to 15 Mbps [8].

End-to-end Delay. Much harder than the pure bandwidth requirements are the delay restrictions that multimedia applications, in particular interactive distributed multimedia applications, impose on communications. The major components that contribute to end-to-end delay are given as follows [9]:

- source compression and packetization delay.
- network transmission delay: including medium access delay (MAC), queuing delay inside the network and propagation delay.
- end-system queuing and synchronization (playout) delay.
- sink decompression, depacketization, and output delay.

Practical experience with multimedia conferencing systems and ITU standards suggests a maximum end-to-end delay of up to 150 msec for interactive video applications

[10]. Uncompressed real-time video applications require a delay bound of 250 msec, but for compressed video, network transmission delay bound should be less than 250 msec because of the encoding and decoding delays along with the queuing and playout delay that contribute to the total delay. A wide range of maximum delay bound can be specified because of the diverse requirements of distributed multimedia applications. For example, to support network-based video games, a response of 50 msec or less is required for twitch actions [11]. On the other hand, for a typical video playback system, initial set-up delay can be in the order of minutes since all frames of the movie will be displayed at constant rate and that will not affect the user's perception of video quality.

Traffic that has an upper bound falls into the class of *synchronous* communication. Most audio-video communication assumes constant delay for all packets² which is called *isochronous* communications. Traffic can be distinguished among the following kinds [9]:

- asynchronous - unrestricted transmission delay,
- synchronous - bounded transmission delay for each message, and
- isochronous - constant transmission delay for each message.

Isochrony does not have to be maintained across the entire path from the source (video encoder) to sink (display device), only at the final destination. A playout buffer is used to recover isochrony due to the variable delay contributed by the network. Some delay variance or jitter of up to 5 msec can be tolerated for practical purposes, but in this

² Here packet corresponds to a voice sample or a video picture.

report, only isochronous VBR traffic is investigated, so the issue of delay variance or delay jitter will not be considered as a separate item.

Reliability Two types of network errors can happen: bit corruption due to noise and packet loss due to congestion in the network [5]. It is expected that packet loss will be more common than bit error due to reliability of fiber as a transmission medium. In general, packet loss rate should be less than 10^{-3} for acceptable video quality. Timeliness is another important constraint for the correct delivery of a packet since for time-critical data, packet retransmission is not suitable when round-trip time is less than the maximum delay bound. Therefore, forward error correction (FEC) techniques must be used to recover lost data. In this report, it is assumed that network provides high reliability such that the packet loss rate is guaranteed to be small.

1.2 MPEG Video Standard

MPEG has been developed for storing video (and associated audio) on digital storage media, which include CD-ROM, digital audio tapes, magnetic disks and writable optical disks, as well for delivering video over networks and telecommunication channels. Compared to other standards, such as CCITT H.120 and H.261, MPEG standard provides better visual quality at higher rates. Phase one of MPEG standard, known as MPEG-1, is not intended to be broadcast television quality so MPEG-2 has been developed to address the compression of television broadcast signals at 10-45 Mbps to provide from VHS-quality to HDTV-quality broadcasting. At the frame level, both MPEG-1 and MPEG-2 generate similar traffic characteristics, therefore only MPEG-1 standard will be described.

The MPEG encoder produces a coded bit stream representing a sequence of encoded pictures from uncompressed video data which is a set of pictures displayed sequentially. The MPEG standard specifies three types of encoded pictures: I (Intracoded), P (Predicted), and B (Bidirectional). Two parameters define the sequence of encoded pictures: M , the distance between I or P pictures, and N , the distance between I pictures. For example, if M and N are given as 3 and 15 respectively, then the sequence of encoded pictures is

I B B P B B P B B P B B P B B . . .

where the pattern IBBPBBPBBPBBPBB repeats indefinitely. Pictures in an MPEG video sequence are organized into groups to facilitate random access. Each group of pictures (GOP) contains the repeating pattern which makes it possible to begin decoding at intermediate points in the video sequence.

MPEG uses an interframe coding technique called motion compensation such that P- and B-frames exploit the temporal redundancy present in a video sequence and are coded with reference to other P- and/or I- frames. P-frames update the picture (using a predictive algorithm) from the last I- or P-frame. B-frames use the bidirectional prediction method and are coded with respect to the preceding I- or P-frame and the subsequent I- or P-frame in the sequence. In general, I-frames are much larger than P-frames, and P-frames are much larger than B-frames. Therefore, an MPEG decoder that compresses a video signal at a constant frame rate (e.g., 30 frames/sec) generates a coded bit stream with a highly variable instantaneous bit rate. In Figure 1.1, an example of MPEG video sequence is shown. Note the rate fluctuations from one picture to another. In some cases,

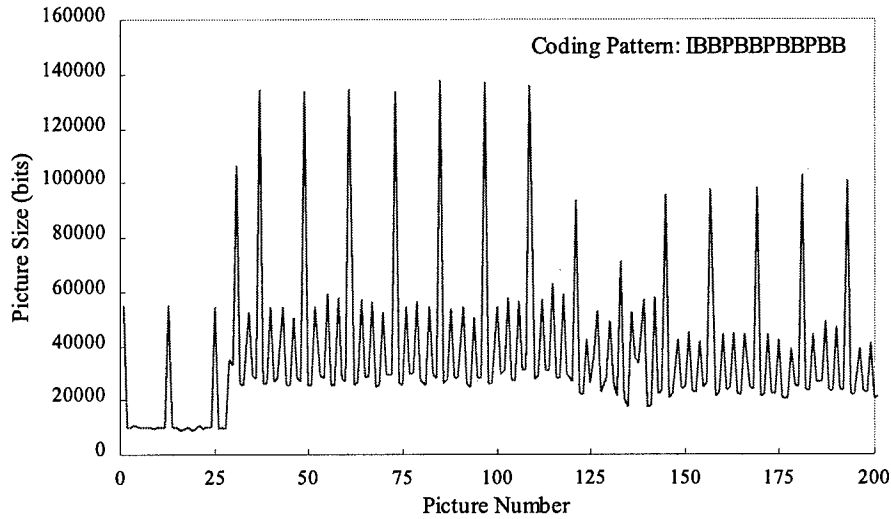


Figure 1.1: An MPEG video sequence.

the fluctuations can be in the order of 10 or more. Consider an I-frame which is 140,000 bits long. Transmitting the I-frame in $1/30$ second over a network would require a transmission capacity of 4.2 Mbps to be allocated. Then during the next $1/30$ seconds, the transmission capacity required for B-frame drops to 0.8 Mbps. These very large rate fluctuations are a consequence of the use of interframe coding techniques in MPEG. Therefore, MPEG constitutes a good example for bursty VBR traffic. However, the encoder output rate, on average, does not change as rapidly as the frame sizes when the scene in the video sequence being encoded changes. Pictures of scenes with more complexity and a lot of motion require more bits to encode. This observation, also pointed in [13], is particularly important since size of the same type of picture in the previous GOP will be used to estimate future traffic when smoothing the MPEG video.

In this report, MPEG video sequences are used to evaluate the performance of the proposed algorithms. The data set consists of sequences of MPEG-1 frame sizes created at the Institute of Computer Science at the University of Würzburg [12]. In all, 19 se-

quences (sportcasts, movies, music videos, newscasts, talk shows, cartoons and sport events) of 40,000 frames each are provided in the data set. In addition, MPEG-1 frame trace of full length Star Wars movie is used in some experiments [60].

1.3 Contributions

Traffic shaping of bursty VBR traffic has been designed and implemented to guarantee conformance of the traffic to the negotiated contract. Most multimedia applications are time-critical and cannot tolerate large delays contributed by the traffic shaping. As described in Section 1.1, multimedia applications have diverse requirements on the end-system in terms of delay and bandwidth guarantees. On the other hand, behavior of the submitted traffic must be as close as possible to the ideal traffic desired by the network to efficiently utilize its resources. New traffic shaping or smoothing³ algorithms must be designed with the following functionality to provide the above requirements:

- 1) A wide set of application constraints must be satisfied. These constraints are usually expressed in terms of maximum smoothing delay and buffer size. Therefore, a unique solution must be provided for all possible set of constraints.
- 2) Smoothing must be optimal in the sense that given a delay or buffer size bound, bursts of data must be spread over the allowed time as much as possible.

³ The term smoothing and traffic shaping will be used interchangeably through the rest of the report since most of time, shaped traffic is smoother than the original traffic because of the allowed smoothing delay or buffer.

- 3) Smoothing of traffic must take into account the underlying network service in order to submit traffic with desired characteristics for maximum network utilization.
- 4) Smoothing must address both stored (off-line) and interactive (real-time) applications.

The report describes the design and specification of a smoothing algorithm that possesses the above listed features. The proposed algorithm is shown to provide a desired traffic behavior for the network given a set of constraints imposed by the application. The smoothing algorithm can be viewed as a bridge closing the gap between the application and the network, each of which has its own specific requirements. The algorithm provides a unique solution for all possible application scenarios by modeling the whole communication system from source to sink and by incorporating the constraints in the system model. Optimality of the algorithm is guaranteed by finding the shortest path through upper and lower bounds on the cumulative rate function that are derived from the given set of constraints. Even with a rudimentary rate prediction rule, the performance of the algorithm is shown to be superior over other algorithms proposed in the literature. The novel idea of choosing the rate based on either past information to minimize traffic variation or future prediction to minimize number of rate changes is presented.

Traffic shaping has a direct impact on bandwidth allocation, therefore traffic shaping must be integrated with bandwidth allocation. This novel idea is presented by proposing a new bandwidth renegotiation algorithm that tracks the bandwidth requirements of the VBR video source using a moderate renegotiation rate which results in

higher bandwidth efficiency. This approach allows for deterministic delay bounds and constant quality video transmission in contrast to other approaches that provides non-deterministic smoothing delay bounds and variable-quality video transmission as a result of shaping the traffic at either encoder buffer or leaky-bucket. In addition, the proposed scheme provides universal interoperability by decoupling the source from the network, thus any application can use the bandwidth renegotiation algorithm.

The effect of traffic shaping on the network utilization is investigated and two theorems are presented regarding the savings in end-to-end delay bound when all sources smooth their traffic. It has been proven that when ideal smoothing (smoothing of stored data or traffic with known future) is applied by all sources, network can support more connections with the same QoS even for the single-hop case which indicates the optimality of the proposed smoothing algorithm.

Finally, a novel concept called *aggregate smoothing* is introduced which integrates multiplexing of multiple video sources with traffic shaping. It is shown that number of rate changes and variation of the aggregate rate are significantly reduced that allows RCBR network service to be a cost-effective solution for real-time traffic when multiple traffic can be transmitted as a bundle. This result is particularly important for public networks carrying aggregated traffic since network utilization is shown to increase with the use of aggregate smoothing.

The remainder of the report is organized as follows. Chapter 2 describes a lossless smoothing algorithm for isochronous VBR traffic given a set of constraints in terms of maximum smoothing delay and buffer size. The performance of the proposed algorithm is

evaluated using MPEG video sequences and is compared to the previous work. In Chapter 3, the effect of smoothing on deterministic end-to-end performance guarantees for packet-switching networks is investigated. Chapter 4 introduces a new concept called aggregate smoothing which integrates traffic shaping with multiplexing of multiple video sources in order to smooth the aggregate rate. A discussion about the effect of aggregate smoothing on network utilization is also provided. Finally, Chapter 5 concludes the report with a summary of the presented work and some research issues for future work using the framework developed in this report.

Chapter 2

Lossless Smoothing Algorithms for VBR Traffic

2.1 Introduction

The transfer of compressed data is demanding in terms of network Quality of Service (QoS) requirements since interframe compression techniques, such as those used in MPEG video, lead to a very bursty bit stream which complicates the problem of network resource management. The effect of smoothing traffic sources on QoS and network utilization has been an important issue in providing end-to-end performance guarantees [43-46]. Several techniques have been developed to control the rate fluctuations of video in order to alleviate congestion and to increase network utilization. Some of these techniques are lossy [32-34], and are inappropriate for smoothing rate fluctuations that are the consequence of interframe compression. The problem of smoothing satisfying a delay bound D , was analyzed by [35], assuming all picture sizes are known a priori and where the selection of rate is designed such that the number of rate changes over time is minimized. A similar algorithm based on the one in [35] that allows specification for two more parameters, K , the number of pictures with known sizes, and, H , a lookahead inter-

val was proposed in [13] to improve algorithm performance. Both algorithms consider picture delay only at the server and no parameter for maximum buffer size is provided. Another work includes three algorithms to smooth VBR MPEG video in the presence of a leaky bucket ATM network access controller [49]. Optimal bandwidth allocation algorithms have been extensively studied in [36] where only stored video is assumed for delivery. In [20] causal and non-causal algorithms are presented that find optimal allocation from a given set of discrete service rates minimizing the total cost, subject to either buffering or delay constraint. The non-causal algorithm has fairly high runtime complexity which becomes impractical when the number of available service rates is large and the enforcement of using a set of discrete service rates is a limitation on the optimality of the solution. Other work consider the case of smoothing in terms of statistical performance guarantees, for example as in [47]. Smoothing is done by periodic averaging of a source's rate (PARing) and large deviations techniques are then used to determine the buffering requirements at the source and the loss probabilities inside the network.

One common feature of the previous approaches is that a solution is optimized for only a specific set of delivery requirements, e.g., either from buffering or delay point of view. This makes it difficult to modify a proposed algorithm for a new set of requirements other than the one it is specified for. Another desired feature, that is lacking in previous work is the ability to incorporate network feedback into the algorithm to adapt to changes in QoS of the underlying network service. This necessitates the use of both server and client buffer status information to be utilized in the algorithm for its adaptive behavior.

In this chapter, causal and non-causal smoothing algorithms for lossless transmission of compressed video data are introduced. Compressed video is bursty in nature due to the interframe compression techniques used by the encoder. A deterministic approach that does not allow the data to be discarded at the end hosts is considered. The causal algorithm is characterized by a set of parameters expressing delivery requirements that include maximum server and client buffer sizes, delay bounds, lookahead interval and number of pictures with known sizes. From the specified parameters, upper and lower bounds on the cumulative transmitted bit rate are derived. Shortest path through these bounds gives the minimum slope, therefore the minimum number of rate changes. Since this solution is non-causal, prediction techniques based on past history should be used to determine the picture sizes in the future for live data applications.

The approach in the design of the proposed algorithm allows it to address a wide range of application scenarios ranging from a live video-conference application with small delay requirement in the order of milliseconds to a video-on-demand application where the delay can be huge in the order of minutes due to large buffers used at the customer site. In the first scenario, the burstiness of the network bandwidth requirement is controlled by delaying data at the encoder buffer, whereas in the second scenario, a pre-fetch buffer at the receiver is filled in advance of each burst by delivering more data across the network than needed, and drained in the course of the burst. The performance of the algorithm has been demonstrated to be effective and in some cases better than other proposed techniques designed and optimized for that particular application requirements.

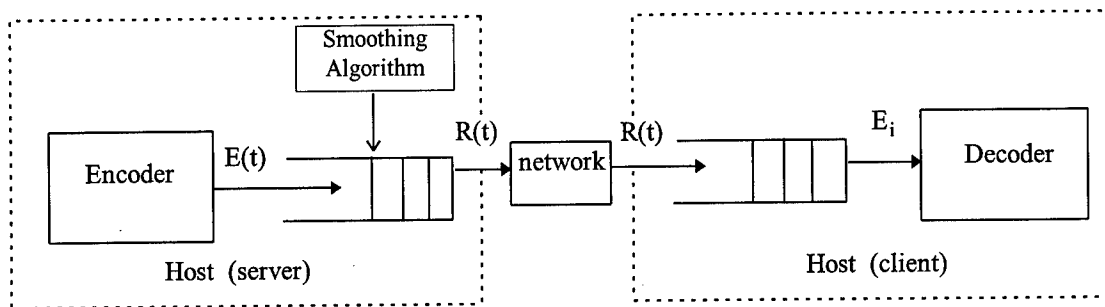


Figure 2.1: System model for smoothing algorithm.

This chapter is organized as follows. In Section 2.2, the system model is introduced and the boundary conditions on the cumulative transmitted bit rate are derived in order to satisfy buffer and delay constraints. In Section 2.3, a causal smoothing algorithm which finds the shortest path through the bounds in order to minimize the number of rate changes is introduced. In Section 2.4, experimental results of the original causal algorithm are presented and an improvement to the algorithm which further minimizes the number of rate changes, is described. The performance of the algorithm is evaluated with respect to other techniques proposed in [13, 35, 48, 49]. The conditions for which buffer or delay constraint should be used to obtain the optimal performance from non-causal algorithm are also specified. Finally, Section 2.5 has some concluding remarks.

2.2 System Model

A video sequence is assumed to be displayed at the rate of $1/T$ pictures per second where T is the picture period. The foundation of the proposed model is based on the work in [37] where the constraints on the output bit rate from the server are derived to ensure

that the server and client buffers do not underflow or overflow. The delivery requirements can be divided into two groups:

- a) delay bound on the pictures at the server and/or client buffer(s).
- b) server and/or client buffer(s) do not overflow and/or underflow.

Although the algorithm described in this chapter is intended to be used for compressed video which is isochronous in nature, any arbitrary bit stream, be it periodic or non-periodic, can utilize the algorithm by dividing time into minimum units of T seconds which indicate the amount of data arrived within that period or in other words, by sampling the input traffic every T seconds. The value of T should be chosen to represent the unit of time which is small enough to bound delay of the traffic that arrives in a period, e.g., for distributed real-time simulation processes requiring bounded delay on each message they interchange with processes in other hosts, T determines the minimum message delay. Through the rest of the report, the term *picture*¹ will be used to denote the arrived data within the period T and the source traffic is assumed to be variable bit rate compressed video. The encoding of the video is open-loop with no rate-control feedback from the smoothing algorithm differing from the techniques that adjust encoder's quantizer scale resulting in lower bit rate at the expense of poorer visual quality [32-34, 50]. The introduced algorithm is lossless in the sense that no data is discarded by the smoothing algorithm and it is assumed that lower layers are responsible for discarding or delaying data, e.g., tagging cells at the user-network interface when smoothed traffic is not

¹ In this dissertation and the literature, the terms frame and picture are often used interchangeably.

conforming to the user specified parameters or discarding cells queued inside the network during the congestion.

The model for rate smoothing is similar to the one described in [39] which consists of two FIFO queues, one with input from the output of an encoder (or storage system) and output to network and other with input from the network and output to decoder (see Figure 2.1). It is assumed that encoding (decoding) time or data transfer time from storage disk to buffer is less than or equal to T seconds. Let $E(t)$ denote the output rate of the encoder at time t and E_i ($i = 1, 2, 3 \dots$), the number of bits in the interval $[(i-1)T, iT)$ referring to the size of picture i and similarly, let $R(t)$ denote the transmitting bit rate at time t and R_i , the number of bits that are transmitted during the interval $[(i-1)T, iT)$. Let $B^e(t)$ and $B^d(t)$ denote the instantaneous fullness (the amount of data in the buffer) of the server and client buffers, respectively. The server buffer receives bits at rate $E(t)$ from the encoder and outputs bits at rate R_i bits per period. The client buffer receives bits at rate $R(t)$ from the network and drains the buffer at rate E_i . $R(t)$ is a piece-wise constant function with possible rate changes occurring periodically at specific locations in time, e.g., at the beginning of the period. This is somewhat different from the approach in [13, 35, 48] which updates the rate that will be used for all of the data belonging to each picture in an aperiodic manner. For each interval with constant rate, it is assumed that packets or cells are scheduled to be sent uniformly spaced in time.

2.2.1 Delay Constraints

End-to-end delay of a picture consists of components such as time spent at the server and client buffers and network access and transmission delays. For live-video applications most of the delay is at the server since the server cannot send bits faster than the encoder can produce them, whereas for stored-data applications, a picture can be pre-fetched at the client buffer well before its display time.

It is assumed that end-to-end delay of a picture i , D^{tot} , is constant and defined as $D^{\text{tot}} = D_i^e + D^{\text{net}} + D_i^d$ where D_i^e and D_i^d are the delays at the server and client buffers respectively and are defined later in this chapter, and D^{net} is the network delay. Let K be the number of pictures in the buffer ready to be sent to the network. The entire picture is assumed to have arrived to the buffer at the beginning of each period before the bits of that picture can be transmitted so $K = 1$ is the minimum number of pictures that must be in the buffer before transmission can begin². If the last bit of picture i is sent at the time s_i , then the delay of the picture at the server buffer is given by

$$D_i^e = s_i + (K + 1)T - iT \quad (2.1)$$

which includes the picture's encoding, queuing and sending delay. A constant delay T , is included in the definition, to denote the upper bound on the encoding delay, defined as the difference between the frame capture time and the time it is fully encoded. Note that in an actual system, the encoding of picture may take less than T , in which case the delay

² This condition is also required to guarantee no violation of delay and buffer constraints as it will be explained later.

of each picture may be smaller than the calculated value using (2.1) but the difference would be negligible.

At the decoder, a new time index τ is defined, which is zero when the first bit arrives at the client and τ_0 as the time when the decoder starts decoding the first picture. The delay of picture i at the client is given by

$$D_i^d = \tau_0 + (i-1)T - a_i \quad (2.2)$$

where a_i is the arrival time of that picture to the client buffer ($a_i = a_{i-1} + s_i - s_{i-1}$) and

$$D_{\max}^d + T \geq \tau_0 \geq a_1.$$

Let D_{\max}^e and D_{\max}^d be the bounds on picture delay at the server and client buffers.

For a given end-to-end delay, the choice of parameter D_{\max}^e determines average buffer occupancy at the server. Because of the conservation of the bits in the transmission pipe, a lower buffer occupancy at the server implies higher buffer occupancy at the client. If D^{tot} is constant, then $D_{\max}^d = D^{\text{tot}} - D^{\text{net}} - (K+1)T$. The choice of τ_0 determines whether data can be prefetched to the client before it is generated at the encoder. For live-video applications, $\tau_0 = D_{\max}^d + T$ must be chosen since the server can not send faster than the encoder and any smaller value results in starvation of the client buffer. However for prerecorded data, smaller value of τ_0 is possible since data can be prefetched at the client buffer. Basically, $D_{\max}^d + T - \tau_0$ determines maximum number of pictures that can be prefetched to the client buffer. The choice of τ_0 should also include network jitter when network delay is not constant. Let D_{\max}^{net} and D_{\min}^{net} be the maximum and minimum network delays. The time when decoding of the first picture starts must include extra

delay of $(D_{\max}^{\text{net}} - D_{\min}^{\text{net}})$ in the case of variable delay. So τ_0 must be chosen as $\tau_0 \geq a_1 + (D_{\max}^{\text{net}} - D_{\min}^{\text{net}})$ in order to compensate for network jitter.

Assuming $T=1$, L^e and L^d are defined as $L^e = \lceil D_{\max}^e - (K+1) \rceil$ and $L^d = \lceil D_{\max}^d - \tau_0 + 1 \rceil$ where $D_{\max}^e \geq (K+1)$ and $D_{\max}^d \geq 0$, $\epsilon^e = L^e - (D_{\max}^e - (K+1))$ and $\epsilon^d = L^d - (D_{\max}^d - \tau_0 + 1)$. At $t=0$, K pictures are available and ready to be transmitted (encoding of the first picture is assumed to start at $t=-K$). Each boundary condition on the cumulative transmitted bit rate is expressed as $\langle x_i, y_i \rangle$ where x_i denotes the time at which boundary condition i applies and y_i for the value of the bound.

Lemma 2.1 The following set of upper and lower bounds on the cumulative transmitted bit rate guarantee that $D_i^e \leq D_{\max}^e$ and $D_i^d \leq D_{\max}^d$:

$$L_D = \left\{ \langle x_i, y_i \rangle : x_i = i - \epsilon^e, y_i = \sum_{j=1}^{i-L^e} E_j \right\} \quad \text{for } L^e < i \leq N,$$

and

$$U_D = \left\{ \langle x_i, y_i \rangle : x_i = i + \epsilon^d, y_i = \sum_{j=1}^{i+L^d} E_j \right\} \quad \text{for } 1 \leq i \leq N - L^d.$$

Proof:

Lower bound: Using (2.1), the last bit of the first picture must be sent before

$t = D_{\max}^e - K$ which requires $\int_0^{D_{\max}^e - K} R(s) ds \geq E_1$ for the transmitted bit rate. For picture i ,

the last bit must be sent at $t = D_{\max}^e + (i - (K+1))$ that can be satisfied if

$$\int_0^{D_{\max}^e + i - (K+1)} R(s) ds \geq \sum_{j=1}^i E_j \quad \text{or} \quad \sum_{j=1}^{i+L^e} R_j - \int_{D_{\max}^e + i - (K+1)}^{i+L^e} R(s) ds \geq \sum_{j=1}^i E_j \quad \text{from which}$$

$$\sum_{j=1}^i R_j - \int_{i-\epsilon^e}^i R(s) ds \geq \sum_{j=1}^{i-L^e} E_j \quad \text{can be obtained providing the lower bound on the cumulative}$$

transmitted bit rate at $t = i - \epsilon^e$.

Upper bound: Using (2.2), arrival time of picture i must satisfy $a_i \geq \tau_0 + (i-1) - D_{\max}^d$.

Since $\int_0^{a_i} R(\tau) d\tau = \sum_{j=1}^i E_j$, then $\int_0^{\tau_0 + (i-1) - D_{\max}^d} R(\tau) d\tau \leq \sum_{j=1}^i E_j$ is obtained. This can be written as

$$\sum_{j=1}^{i-L^d} R_j + \int_{i-L^d}^{\tau_0 + (i-1) - D_{\max}^d} R(\tau) d\tau \leq \sum_{j=1}^i E_j \quad \text{or} \quad \sum_{j=1}^i R_j + \int_i^{i+L^d} R(\tau) d\tau \leq \sum_{j=1}^{i+L^d} E_j \quad \text{which provides the upper}$$

bound on the cumulative transmitted bit rate at $\tau = i + \epsilon^d$.

When D_{\max}^e and $(D_{\max}^d - \tau_0)$ are integer values, $x_i = i$ is obtained for both upper and lower bounds which gives:

$$\sum_{j=1}^{i-L^e} E_j \leq \sum_{j=1}^i R_j \leq \sum_{j=1}^{i+L^d} E_j \quad (2.3)$$

for $L^e < i \leq N - L^d$.

It should be noted that when $L^d = 0$ and $L^e > 0$, server is not allowed to send faster than the encoder and in the case of $L^e = 0$ and $L^d > 0$ data is prefetched to the client buffer.

2.2.2 Buffer Constraints

The following conditions must be met in order to ensure sender and client buffers do not overflow or underflow:

$$B_{\min}^e \leq B^e(t) \leq B_{\max}^e \quad \forall t \quad \text{and} \quad 0 \leq B^d(t) \leq B_{\max}^d \quad \forall t. \quad (2.4)$$

where B_{\max}^e and B_{\min}^e denote the maximum and minimum server buffer sizes respectively, and B_{\max}^d denotes the maximum client buffer size. It should be noted that B_{\min}^e can be negative which indicates the amount of data that can be prefetched to the client buffer. In the case of network jitter, the extra buffer at the client is also required since additional bits may arrive with shorter delay. For safety margin, $B_{\max}^d - \max_{t \geq 0} \int_t^{t+D_{\max}^{\text{net}}-D_{\min}^{\text{net}}} R(s) ds$ must be used as the maximum client buffer size when computing the bounds.

Assuming that there are K pictures available in the buffer at $t = 0$, buffer occupancy at the server after encoding picture i is, $B_i^e = B^e(i) = \int_0^i [E(s+K) - R(s)] ds + \sum_{j=1}^K E_j$ which can be written as

$$B_i^e = \sum_{j=1}^K E_j + \sum_{j=1}^i E_{j+K} - \sum_{j=1}^i R_j \quad (2.5)$$

at $t = i$ when $i \geq 1$.

For the client buffer, the time index τ is used which is defined as $t = \tau + \text{network delay}$ ($\tau = 0$ when first bit arrives at the client). At time τ_0 , decoding of the first picture starts. Let $L = \lceil \tau_0 \rceil$. The client buffer fullness when $\tau = \tau_0$ is given by

$B_0^d = \sum_{j=1}^{L-1} R_j + \int_{(L-1)}^{\tau_0} R(s) ds$. The client buffer fullness at time $\tau = i + \tau_0$ is then given by

$B_i^d = B_{i-1}^d + \int_{i+\tau_0-1}^{i+\tau_0} R(\tau) d\tau - E_i$. If this expression is rewritten recursively and i is substituted

with $i + L$, the following is obtained:

$$B_{i-L}^d = \sum_{j=1}^{i-1} R_j + \int_{i-1}^{i-L+\tau_0} R(\tau) d\tau - \sum_{j=1}^{i-L} E_j \quad (2.6)$$

at $\tau = i - L + \tau_0$ when $i > L$.

The following lemma provides the set of upper and lower bounds for buffer constraint.

Lemma 2.2 The following set of upper and lower bounds on the cumulative transmitted

bit rate guarantees that $B_{\min}^e \leq B^e(t) \leq B_{\max}^e \quad \forall t$ and $0 \leq B^d(t) \leq B_{\max}^d \quad \forall t$:

$$L_B^S = \left\{ \langle x_i, y_i \rangle : x_i = i, y_i = \sum_{j=1}^K E_j + \sum_{j=1}^i E_{j+K} - B_{\max}^e \right\} \quad \text{for } i \geq 1,$$

$$L_B^C = \left\{ \langle x_i, y_i \rangle : x_i = i - L + \tau_0, y_i = \sum_{j=1}^{i-L} E_j \right\} \quad \text{for } i > L,$$

$$U_B^S = \left\{ \langle x_i, y_i \rangle : x_i = i, y_i = \sum_{j=1}^K E_j + \sum_{j=1}^i E_{j+K} - B_{\min}^e \right\} \quad \text{for } i \geq 1,$$

$$U_B^C = \left\{ \langle x_i, y_i \rangle : x_i = i - L + \tau_0, y_i = \sum_{j=1}^{i-L} E_j + B_{\max}^d \right\} \quad \text{for } i > L.$$

Proof:

Using (2.4), (2.5) and (2.6), the following relations can be derived to guarantee that neither the server nor client buffers overflow or underflow :

$$\sum_{j=1}^K E_j + \sum_{j=1}^i E_{j+K} - B_{\max}^e \leq \sum_{j=1}^i R_j \leq \sum_{j=1}^K E_j + \sum_{j=1}^i E_{j+K} - B_{\min}^e \quad \text{at } t = i \text{ when } i \geq 1$$

and

$$\sum_{j=1}^{i-L} E_j \leq \sum_{j=1}^{i-1} R_j + \int_{i-1}^{i-L+\tau_0} R(\tau) d\tau \leq \sum_{j=1}^{i-L} E_j + B_{\max}^d \quad \text{at } t = i - L + \tau_0 \text{ when } i > L.$$

The rest of the proof is obvious.

Finally $U = U_B^S \cup U_B^C \cup U_D$ and $L = L_B^S \cup L_B^C \cup L_D$ provide the set of upper and lower bounds for the given buffer and delay constraints. A special case is when D_{\max}^e , $(D_{\max}^d - \tau_0)$ and τ_0 are integers aligning all the bounds at $x_i = i$. Then the bounds can be expressed as:

$$L_i \leq \sum_{j=1}^i R_j \leq U_i, \quad (2.7)$$

where

$$L_i = \max \left\{ \sum_{j=1}^{i-L^e} E_j, \sum_{j=1}^K E_j + \sum_{j=1}^i E_{j+K} - B_{\max}^e, \sum_{j=1}^{i-L} E_j \right\}$$

and

$$U_i = \min \left\{ \sum_{j=1}^{i+L^d} E_j, \sum_{j=1}^K E_j + \sum_{j=1}^i E_{j+K} - B_{\min}^e, \sum_{j=1}^{i-L} E_j + B_{\max}^d \right\}.$$

2.3 Smoothing Algorithm

In this section, it is described how to find the optimal path through the derived upper and lower bounds using the shortest path algorithm. First, the shortest path algorithm which is the key to the smoothing algorithm is introduced. Then a causal smoothing algorithm specified by the following parameters is introduced: delay bounds and buffer sizes at the server and client, the number of pictures with known sizes and look-ahead interval for improving the algorithm performance.

2.3.1 Shortest Path

In Section 2.2, necessary upper and lower bounds on the cumulative transmitted bit rate were derived. The purpose of the rate smoothing algorithm is to find an optimal path through the bounds such that a minimum number of rate changes is obtained. Since the slope of a path through the bounds gives the instantaneous rate, optimal path should have the minimum slope and the shortest length to minimize number of rate changes. Furthermore, this path will minimize the maximum rate since it will have the smallest possible slope. The shortest path algorithm described in this section is based on the work in [38] where the problem of constructing a Euclidean shortest path between two specified points, which avoids a given set of barriers, is addressed.

The shortest path algorithm uses two chains branching at some vertex, called a *cusp*. One chain includes a set of vertices, U_i , $i = 1, 2, \dots, N$, each of which belongs to the set of upper bounds and the other chain includes vertices, L_i , from the set of lower bounds. The source and destination points are specified as s and t respectively. $D(v_i, v_j)$

is defined as the shortest path from v_i to v_j and D_i as the union of the two chains; i.e., $D_i = D(s, U_i) \cup D(s, L_i)$. In [38], it is shown that both $D(s, U_i)$ and $D(s, L_i)$ must be inward-convex polygonal chains; i.e., it is convex with convexity facing toward the interior of the area bounded by the upper and lower boundary points. The algorithm successively constructs D_1, D_2, \dots, D_N and finally $D(s, t)$ ensuring the inward-convexity of the upper and lower chains. In detail:

1. The algorithm constructs D_1 by connecting s to U_1 and L_1 , $s_1 = s$, $k = 1$.
2. General Step (Construct D_{i+1} from D_i): The algorithm first creates $D(s_k, U_{i+1})$ (or $D(s_k, L_{i+1})$) by scanning all vertices in the $D(s_k, U_i)$ beginning from U_i . Two cases are distinguished:
 - (i) a vertex U_j ($1 \leq j \leq i$) is found in $D(s_k, U_i)$ such that when connected with U_{i+1} , the inward convexity is not destroyed. All other vertices between U_j and U_{i+1} are deleted if any.
 - (ii) if no such a vertex is found in $D(s_k, U_i)$, then $D(s_k, L_i)$ is scanned until one vertex, L_m ($1 \leq m \leq i$), is found such that $\overline{L_m U_{i+1}}$ is inside the polygon and $s_{k+1} = L_m$. $D(L_m, U_{i+1})$ and $D(L_m, L_i)$ become the new upper and lower chains. $D(s_k, L_m)$ gives the shortest path between s_k and L_m and k is also incremented; $k = k + 1$.

The general step is applied to $D(s_k, L_i)$ to construct $D(s_k, L_{i+1})$.

3. Final Step: Once D_N has been constructed, the general step is applied to construct $D(s_{k_{\max}}, t)$. The chain $\overline{ss_2 \dots s_{k_{\max}} t}$ gives the shortest path between s and t .

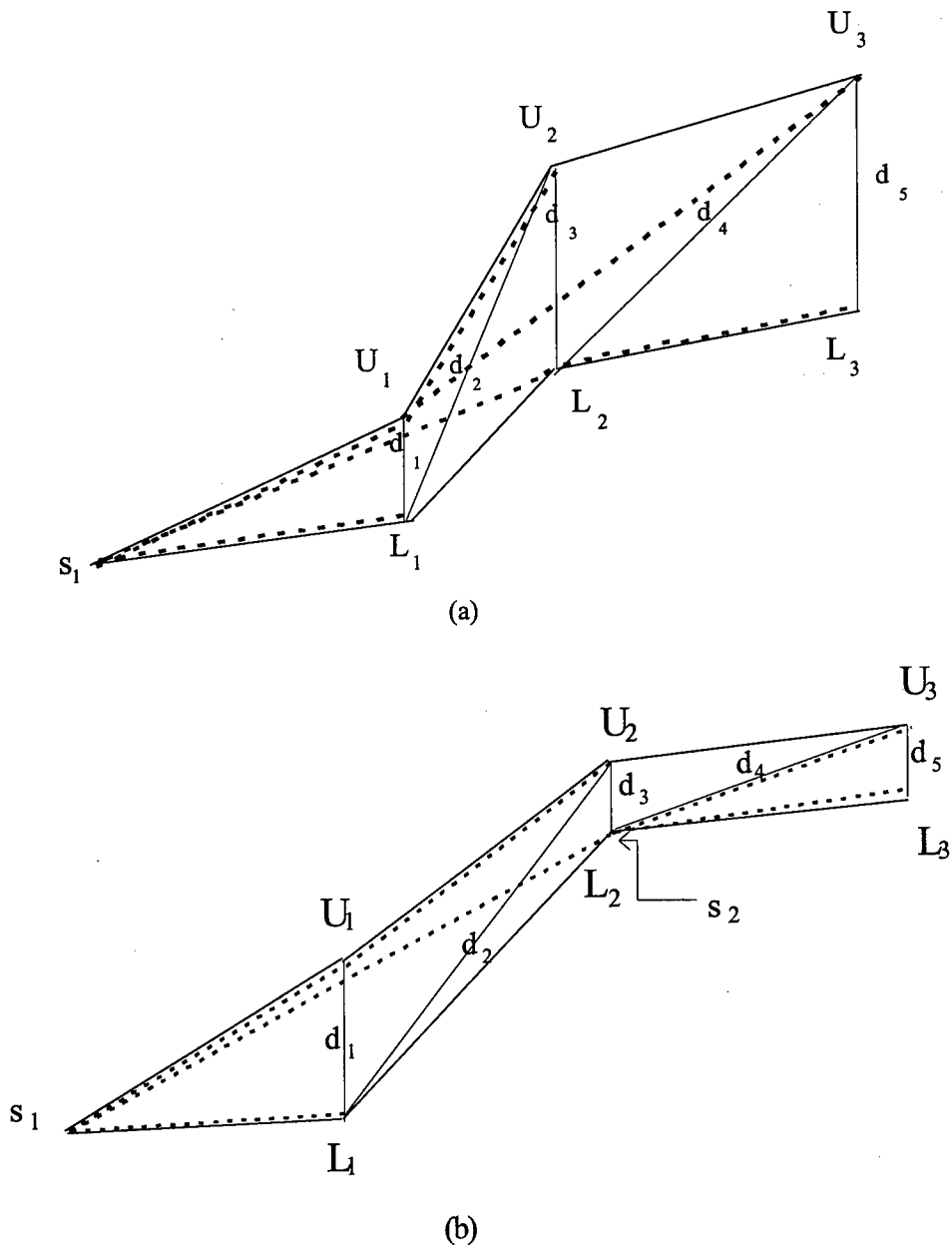


Figure 2.2: Illustration of the shortest path algorithm. In (a) two chains are preserved and in (b) a new cusp is found.

The details and proof of the algorithm can be found in [38]. In Figure 2.2, the shortest path algorithm is illustrated where d_i is the diagonal between the two extreme points of upper and lower chains. Figure 2.2(a) corresponds to case (i) of the general step where upper and lower chains are preserved. The chains are constructed in the order of $\overline{s_1 U_1}$, $\overline{s_1 L_1}$, $\overline{s_1 U_1 U_2}$, $\overline{s_1 L_2}$, $\overline{s_1 U_3}$ and $\overline{s_1 L_2 L_3}$. Figure 2.2(b) illustrates the case (ii) of the general step where a new cusp is found. First, $\overline{s_1 U_1}$, $\overline{s_1 L_1}$, $\overline{s_1 U_1 U_2}$ and $\overline{s_1 L_2}$ are constructed. However, there is no vertex in $\overline{s_1 U_1 U_2}$ which can keep the inward-convexity of the upper chain when connected with U_3 so L_2 becomes the new cusp, s_2 . $\overline{s_2 U_3}$ and $\overline{s_2 L_3}$ become the new upper and lower chains respectively.

The running time of the algorithm for N upper and lower bounds besides s and t is analyzed as follows: case (i) of the general step takes constant time on the average since at one extreme, $U_j = U_i$ for $\forall i$ which takes one comparison at every step (in total N comparisons) and at the other extreme $U_j = s_k$ for $\forall i$ and $\forall k$ which takes two comparisons (in total $2N$ comparisons). Case (ii) may involve scanning a large number of vertices; however once a vertex has been scanned and the corresponding angle has been found to require continuation of the scanning process, that vertex is eliminated from consideration since it belongs to the shortest path whether it becomes the new cusp or not. So the algorithm runs in time $O(N)$.

2.3.2 Causal Algorithm Design and Specification

The shortest path determines noncausally a path through the bounds. For stored-data applications, smoothing is realized by simply applying the shortest path algorithm to the pre-computed bounds using (2.7). However, this solution can not be implemented for live video applications since the size of future pictures is not known. In this case,

```

Smooth (int H, int K, int N, int  $D^e$ , int  $D^d$ , int  $B_{max}^e$ , int  $B_{min}^e$ , int  $B_{max}^d$ , int  $\tau_0$ , int *pic_size) {
    int i, up_d=0; low_d=0, up_b_enc, low_b_enc, up_b_dec=0, low_b_dec=0;
    int temp_up_d, temp_low_d, temp_up_b_enc;
    int temp_low_b_enc, temp_up_b_dec, temp_low_b_dec;
    int  $L^d$ ,  $L^e$ , L, init_buf=0;
    float rate, sum=0;
    coordinate upper_bound[H+1], lower_bound[H+2];

     $L^d = D^d - \tau_0 + 1$ ;
     $L^e = D^e - (K+1)$ ;
    L =  $\tau_0$ ;
    for(i=1; i<K+1; i++)
        init_buf+=size(i,i);
    up_b_enc = init_buf -  $B_{min}^e$ ;
    low_b_enc = init_buf -  $B_{max}^e$ ;
    up_b_dec =  $B_{max}^d$ ;
    rate = pic_size[1]/(  $L^e + L^d$  ); /* choose an initial rate */
    for (i=1; i<(N+  $L^e + L^d$  ); i++) {
        upper_bound[0].x = lower_bound[0].x = 0; /* lower_bound[0] and upper_bound[0]
        upper_bound[0].y = lower_bound[0].y = sum; /* includes source point */
        lower_bound[H+2].x = H+1; /* destination is at lower_bound[H+1] */
        lower_bound[H+2].y = source.y + (H+1)*rate; /* rate should not change if possible */
        temp_up_d = up_d; temp_up_b_enc = up_b_enc; temp_up_b_dec=up_b_dec;
        temp_low_d = low_d; temp_low_b_enc = low_b_enc; temp_low_b_dec = low_b_dec;
        for (j=0; j<H, j++) {
            upper_bound[j].x = lower_bound[j].x = j+1;
            upper_bound[j].y = min( temp_up_d+=size(i+j+  $L^d$ , i+K), temp_up_b_enc+=
                                size(i+j+K, i+K), temp_up_b_dec+=size(i+j-L, i+K) );
            lower_bound[j].y = max( temp_low_d+=size(i+j-  $L^e$ , i+K), temp_low_b_enc+=
                                size(i+j+K, i+K), temp_low_b_dec+=size(i+j-L, i+K) );
        }
        rate = find_shortest_path (upper_bound, lower_bound, H);
        notify(rate, i); /* notify server the rate for period i */
        sum = sum+rate;
        up_d = up_d+pic_size[i+  $L^d$  ]; low_d = low_d + pic_size[i-  $L^e$  ];
        up_b_enc = up_b_enc+pic_size[i+K]; low_b_enc = low_b_enc + pic_size[i+K];
        up_b_dec = up_b_dec+pic_size[i-L]; low_b_dec = low_b_dec+pic_size[i-L];
    }
}

```

Figure 2.3: Specification of the causal smoothing algorithm.

bounds are derived using estimates of future picture sizes. The rate prediction problem has been extensively studied from several different perspectives in the past which include using linear (Kalman) prediction [40], an artificial neural network based approach [25], using AR(1)-based bandwidth estimator [20] and other promising methods as described in [41]. The previous work showed that a very simple prediction rule using the past information leads to satisfactory performance for the purpose of smoothing [13, 35]. Especially for large delay bounds, smoothing is not very sensitive to the estimation error resulting from imperfect forecasting. The causal algorithm is designed independently from the estimation rule. Since any estimation method is most effective for pictures in the near future, only the bounds within a look-ahead window are computed using the picture size estimates. The causal algorithm is designed with a parameter H which specifies the size of the look-ahead window. The optimal value of H depends on the estimation method and source bit rate characteristics. For $x_i = i$, the look-ahead window is shifted by a picture period at every step and possible rate changes occur only at the beginning of period. For non-integer case of x_i , step size is the time difference between two consecutive boundary points and rate changes occur only at x_i . The condition $K \geq 1$ that there must be at least one picture with known size within the look-ahead window is imposed since the sequence of upper and lower bounds using the estimates of picture sizes may violate delay and buffer constraints in the case of large estimation errors. By ensuring that the size of the current picture is known, it is guaranteed that the computed smoothed rate will not violate the bounds.

In Figure 2.3, a description of the causal algorithm is presented for input parameters with integer values. It is assumed that $T = 1$. The parameters to the smoothing function include H for look-ahead window size, K for the number of pictures with known sizes in the server buffer, N for the number of pictures to be smoothed, D^e for the delay bound at the server, D^d for the delay bound at the client, B_{\max}^e and B_{\min}^e for the maximum and minimum buffer sizes at the server, B_{\max}^d for the maximum buffer size at the client and τ_0 for the time when decoding starts. There are two functions in the specification: $size(i,j)$ which returns, at period j , either the actual size of picture i , or an estimated size depending on parameter K for $1 \leq i \leq N$, otherwise zero is returned. The function $find_shortest_path()$ finds the shortest path through the bounds in the look-ahead window and returns the slope of the first edge of the shortest path as the smoothing rate for period i . The selection of the destination point is designed to minimize the number of rate changes over time by choosing the rate in the previous period and keeping it as constant in the look-ahead window.

In Figure 2.4, a description of the shortest path algorithm is presented. The smoothing rate is defined as $alpha \cdot min + (1 - alpha) \cdot max$ where min is the rate on the lower path and max on the upper path. There are two conditions which can terminate the algorithm: either a new cusp is found or the destination point is arrived in which case both paths are valid. When a new cusp is found, the shortest path belongs to the other path so $alpha = 1$ if the cusp belongs to the lower path and vice versa. If the destination point is reached (lower path connects the source to the destination point), $alpha = 1$ (lower path) is chosen. In the next section, it will be described how to improve


```

int find_shortest_path (coordinate *upper_bound, coordinate *lower_bound, int H) {

    int up_index[H+1], down_index[H+2];          /* indexes of the bounds forming each path */
    int num_up=2; num_down=2;                    /* number of bounds at each path */
    int exit_flag=1;                             /* exit condition, default=1 when destination can be reached by both
                                                paths, exit_flag=0 when a new vertex is found in the look-ahead window */
    double min, max;                             /* minimum and maximum values of the rate which satisfies the bounds */
    double alpha;                                /* rate = alpha*min + (1-alpha)*max */

    up_index[0] = down_index[0] = 0;              /* source is the vertex */
    up_index[1] = down_index[1] = 1;              /* first step of the shortest-path algorithm */

    for(i=2; i<H+2; i++) {                        /* general step */

        for(k=num_down-1; k>-1; k--)
            if(check_for_convex(k, i, 0)) {         /* check whether lower_bound[i] can be
                                                    connected to lower_bound[down_index[k]] */
                down_index[k+1]=i; num_down = k+2; goto upper_loop;
                /* append lower_bound[i] to lower path */
            }
        for(k=1; k<num_up; k++)                    /* search upper path for a new vertex */
            if(check_for_bound(k,i,0)) {           /* check whether upper_bound[up_index[k]]
                                                    can be a new vertex */
                exit_flag=0; alpha=0; goto finish; /* possible modification here */
            }
        upper_loop:
        if(i<H+1) {                                /* destination point is already reached if i=H+2 by lower path */
            for(k=num_up-1; k>-1; k--)
                if(check_for_convex(k, i, 1)) {     /* check whether upper_bound[i] can be
                                                    connected to upper_bound[up_index[k]] */
                    up_index[k+1]=i; num_up = k+2; goto finish_loop;
                    /* append upper_bound[i] to upper path */
                }
            for(k=1; k<num_down; k++)                /* search lower path for a new vertex */
                if(check_for_bound(k,i,1)) {         /* check whether lower_bound[down_index[k]]
                                                    can be a new vertex */
                    exit_flag=0; alpha=1; goto finish; /* possible modification here */
                }
            }
        finish_loop:
    }

    finish:
    min = (lower_bound[down_index[1]].y-lower_bound[0].y)/
          (lower_bound[down_index[1]].x-lower_bound[0].x);
    max = (upper_bound[up_index[1]].y-upper_bound[0].y)/(upper_bound[up_index[1]].x-upper_bound[0].x);
    if(exit_flag) alpha = 1; /* choose the minimum rate if both paths are valid */

    return (alpha*min + (1-alpha)*max);
}

```

Figure 2.4: Specification of the shortest-path algorithm.

the performance of the algorithm by choosing *alpha* such that the number of rate changes is minimized. Since the algorithm takes constant time $O(H)$ at every step and H is usually chosen to be a small number, it can be easily implemented in real time.

Although the sequence of upper and lower bounds can be pre-computed ahead of the transmission for stored-data applications, in an actual system implementation, the algorithm behavior should be adaptive in order to respond to changes in the buffer and delay constraints because of the change in network QoS guarantees and application requirements. For example, in fully-interactive VOD systems, most video sequences to be transmitted are computed in real-time to support operations such as scanning and editing. For networks without any QoS guarantees, feedback mechanisms can be used to change the parameters of the smoothing algorithm to provide adaptive behavior. In such cases, causal algorithm should be used since future traffic is not deterministic and the sequence of upper and lower bounds must be re-computed.

2.4 Experiments

In this section, the experimental results of the smoothing algorithm are provided to show that the proposed scheme is effective in smoothing bursty traffic under the given set of delay and buffer constraints. A large number of experiments using four MPEG video sequences are performed to evaluate the performance of the causal algorithm. First the effect of the parameters on the original causal algorithm is examined. Then an improvement to the original algorithm is described to further decrease the number of rate changes. In order to justify the proposed algorithm, the existing smoothing algorithms in the literature are summarized and two of the promising algorithms are applied to the same set of MPEG video sequences. The results indicate that the proposed method provides better rate smoothing, especially for bursty traffic with rapidly changing picture sizes.

Finally, the conditions for which buffer or delay constraint should be used when smoothing prerecorded source traffic are discussed.

2.4.1 Performance of Original Causal Algorithm

Live-video applications require buffering at the server for smoothing since the server cannot send faster than the encoder can produce. To evaluate the performance of the causal algorithm, delay constraints $D_{\max}^e = D + T$ and $D_{\max}^d = 0$ seconds are used with no buffer constraints. The choice of client or server for buffering has no effect on the performance since from (2.3), it is seen that as long as $L^e + L^d$ is a constant, the smoothed rate functions for various values of L^e and L^d are almost identical except with a phase shift in time because of the fact that the difference between a lower bound and its corresponding upper bound is always the same. This indicates that the choice of a delay value at the server or client depends on other factors such as cost of the implementation and availability of resources rather than the performance of the smoothing algorithm.

Four MPEG video bit streams are used in the experiments, each of which has 40,000 frames per sequence (about 22 minutes of video for 30 frames/sec frame rate) with encoded picture parameters of $N = 12$ and $M = 3$ [12]. The video bit streams include two movies, "Star Wars" and "Terminator II", a cartoon, "Asterix", and a sports event, "Formula 1 Race: GP Hockenheim 1994". Table 2.1 shows the compression rates and basic statistics of the video sequences used in the experiments. As shown in Figure 2.5, Formula 1 contains many rapid movements and scene changes so its trace shows very large changes in any type of frames, and the B frames are often the same size as the P

Table 2.1: Statistics of the MPEG video sequences used in the experiments.

| Sequence | Comp. rate | Frames Mean [bits] | Frames CoV | Frames Peak/ Mean | GOPs Mean [bits] | GOPs CoV | GOPs Peak/ Mean |
|--------------|------------|--------------------|------------|-------------------|------------------|----------|-----------------|
| Asterix | 119 | 22,348 | 0.90 | 6.6 | 268,282 | 0.47 | 4.0 |
| Formula 1 | 86 | 30,749 | 0.69 | 6.6 | 369,006 | 0.38 | 3.6 |
| Star Wars | 130 | 15,599 | 1.16 | 11.9 | 187,185 | 0.39 | 5.0 |
| Terminator 2 | 243 | 10,904 | 0.93 | 7.3 | 130,865 | 0.35 | 0.74 |

frames. Star Wars and Terminator 2 show the behavior of typical movie sequences with high compression rates because of the slow changing scenes compared to those in the sports events.

In the experiments, the size of picture i , if not known, is estimated to be E_{i-N} which uses the fact that the pictures i and $i-N$ are of the same type (I, B or P). Figure 2.6 shows original bit rate as a function of time for the first 200 pictures of Formula 1 video sequence and three values of the delay bound $D = 0.067, 0.1333$ and 0.2 seconds. The algorithm is run with parameter values of $H = 6$ and $K = 1$. There is still some burstiness associated with the smoothed rate for both causal and non-causal algorithms when $D = 0.067$ second, but for $D = 0.133$ second, the non-causal algorithm gives a smoothed rate function, whereas the causal algorithm output is still bursty compared to that of non-causal algorithm. And for $D = 0.2$ second, both algorithms give very smooth output rate functions. Therefore $D = 0.2$ second is an optimal parameter value to use for causal algorithm if a delay up to 0.2 second is allowed, and $D = 0.133$ second is already an excellent value for non-causal algorithm.

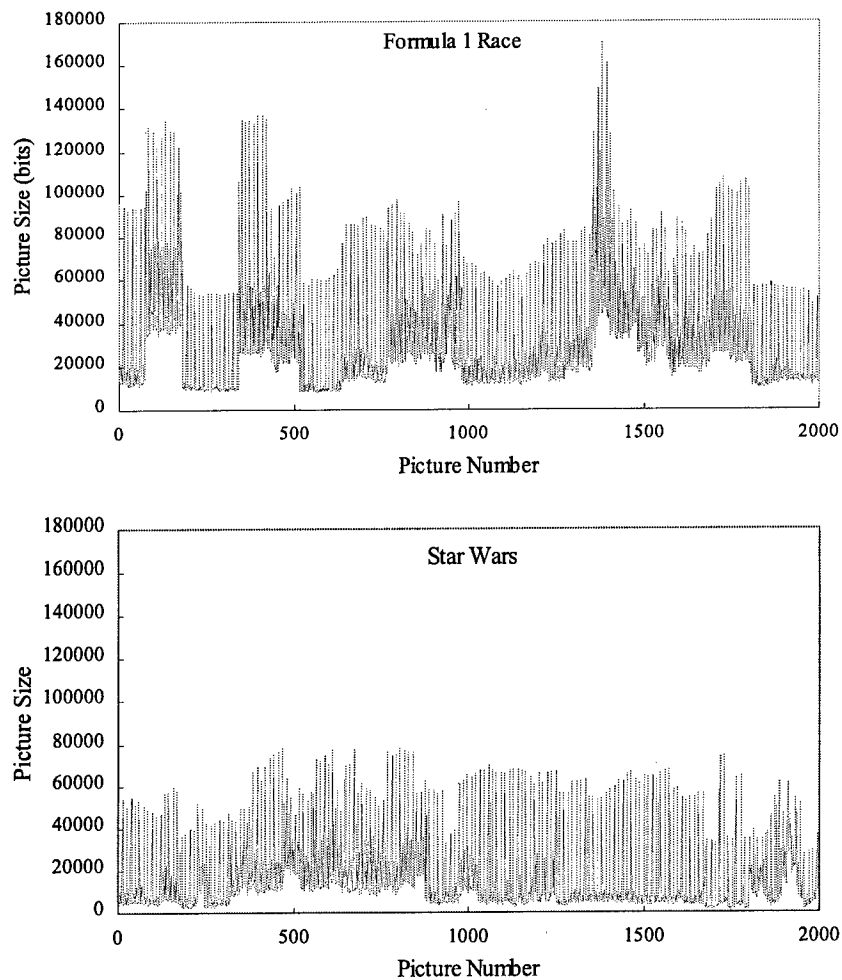


Figure 2.5: Two MPEG video sequences: Formula 1 and Star Wars.

Figure 2.7 shows the delays of individual pictures for two comparisons. In the upper graph, three delay bounds of the causal algorithm are compared. As shown, the delays of pictures are bounded by 0.067, 0.133 and 0.2 second as specified for the casual algorithm. No delay bound violation has been observed in any of the experiments since $K \geq 1$ for all cases. In the lower graph, the picture delays of causal and non-causal algorithms for $D = 0.2$ second are compared. It is observed that when the input rate increases rapidly, the causal algorithm underestimates the picture sizes resulting in larger picture delays, and when the input rate decreases rapidly, the picture sizes are overestimated re-

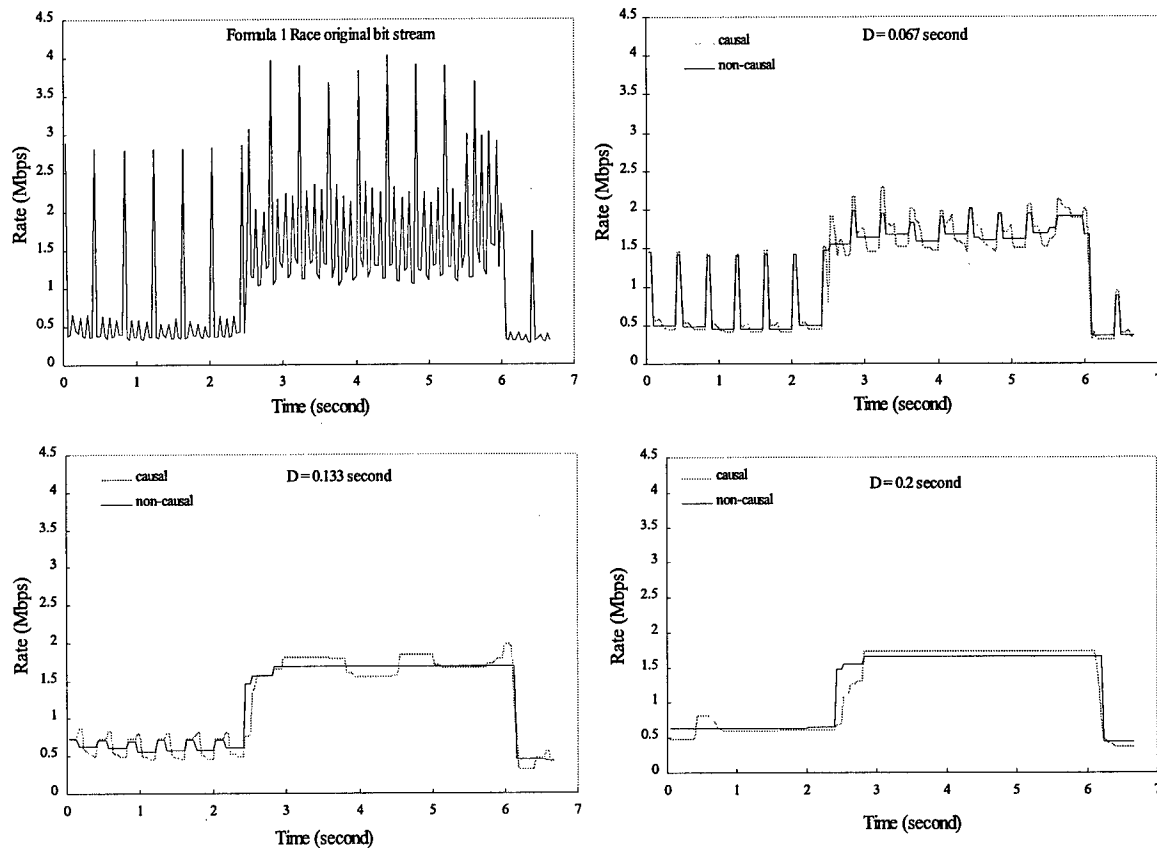


Figure 2.6: Rate as a function of time for original bit stream and three delay bounds.

sulting in smaller picture delays compared to those of the non-causal algorithm.

Different quantitative measures can be defined to characterize the effectiveness of smoothing [13]. Three of them are used to study algorithm performance, as each of the parameters, D , H and K varies. The measures are:

- the number of times the rate function is changed by the algorithm over N periods.
- the maximum value of the rate function over N periods.
- the standard deviation (S.D) of the rate function over N periods.

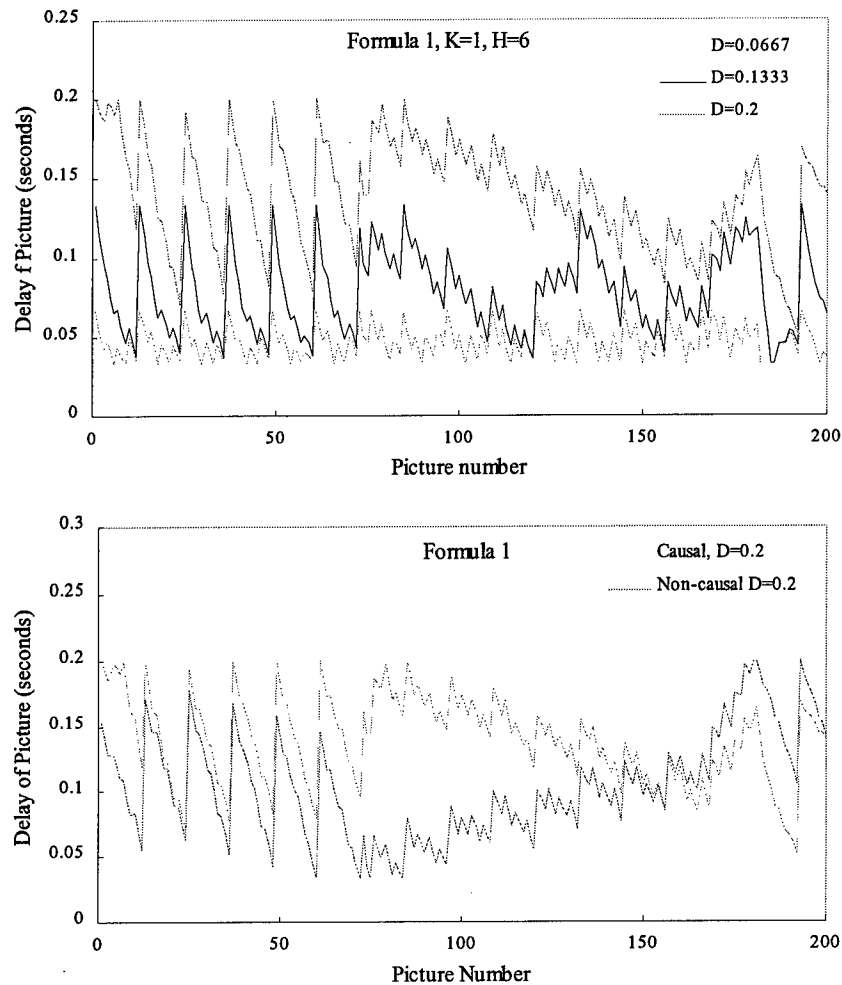


Figure 2.7: Delays of pictures in Formula 1 video sequence.

Figure 2.8 shows the three quantitative measures as a function of delay bound for the four MPEG video sequences when the causal algorithm is applied as specified in Figure 2.3. As it is expected, when delay bound is increased, the rate function becomes smoother. The maximum rate decreases rapidly at first, and then stays almost the same after $D = 0.2$ second and standard deviation of rate behaves similarly for all cases. An interesting observation is that the smoothing effect on the number of rate changes for Formula 1 is larger than those of Star Wars and Terminator 2 which can be explained as follows: when the estimation error is large, the rate function tends to oscillate between

maximum or minimum values resulting in constant rates lasting longer whereas, for other cases, many rate changes are needed to track the input traffic. That is why the standard deviation of rate for Formula 1 is also larger.

Figure 2.9 shows the quantitative measures as a function of the look-ahead interval, H , for the four MPEG video sequences. There is no advantage of using large values of H since the number of rate changes increases as H increases. The standard deviation of rate and the maximum rate do not show any noticeable improvement for values of H larger than 5.

K should be as small as possible to reduce picture delay at the server. The experimental results in Figure 2.10 show that K does not effect the standard deviation of rate and the maximum rate significantly, but the number of rate changes is a decreasing function of K . It is also observed that for $K \geq 5$, all four rate functions have similar number of rate changes which indicate that the smoothing algorithm has the same degree of effect on MPEG encoded bit streams with different statistics when the future is known (assuming the bit streams are encoded with the same parameters, e.g., M and N are the same). A common result of all three set of experiments is that number of rate changes is more sensitive to estimation error than the other two measures.

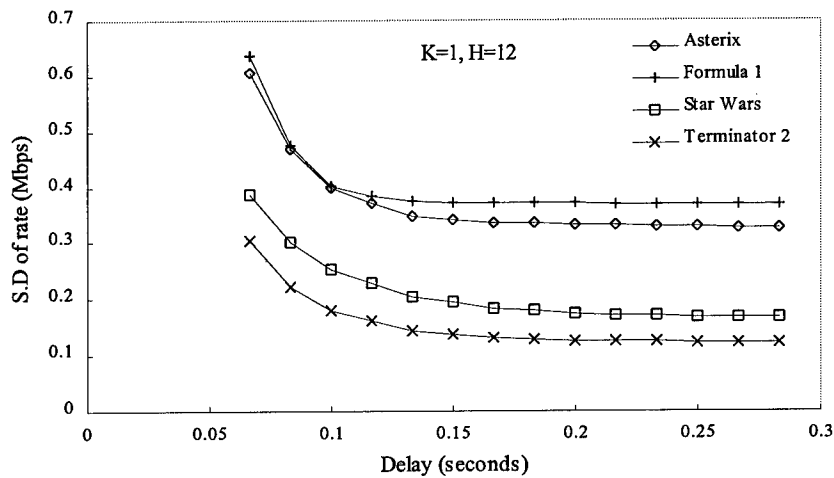
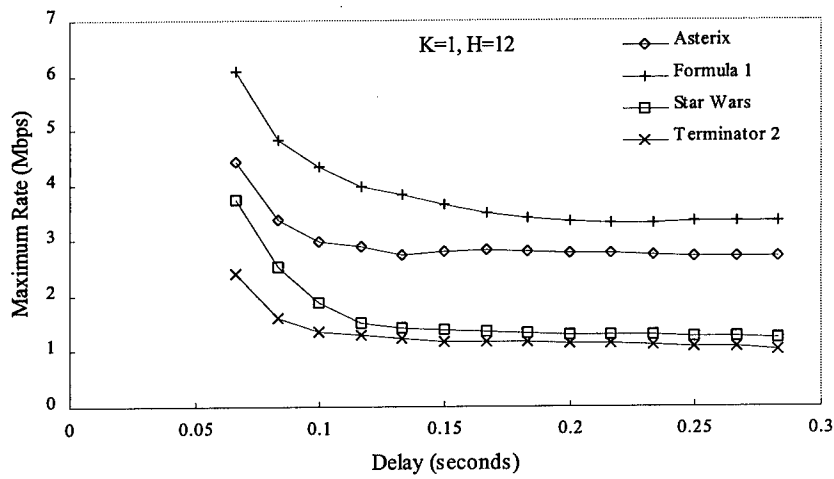
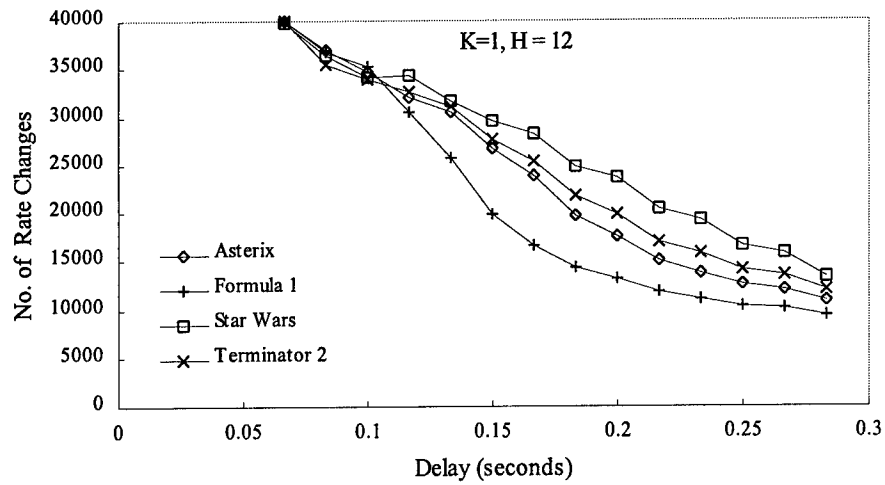


Figure 2.8: Performance of causal algorithm as a function of delay bound.

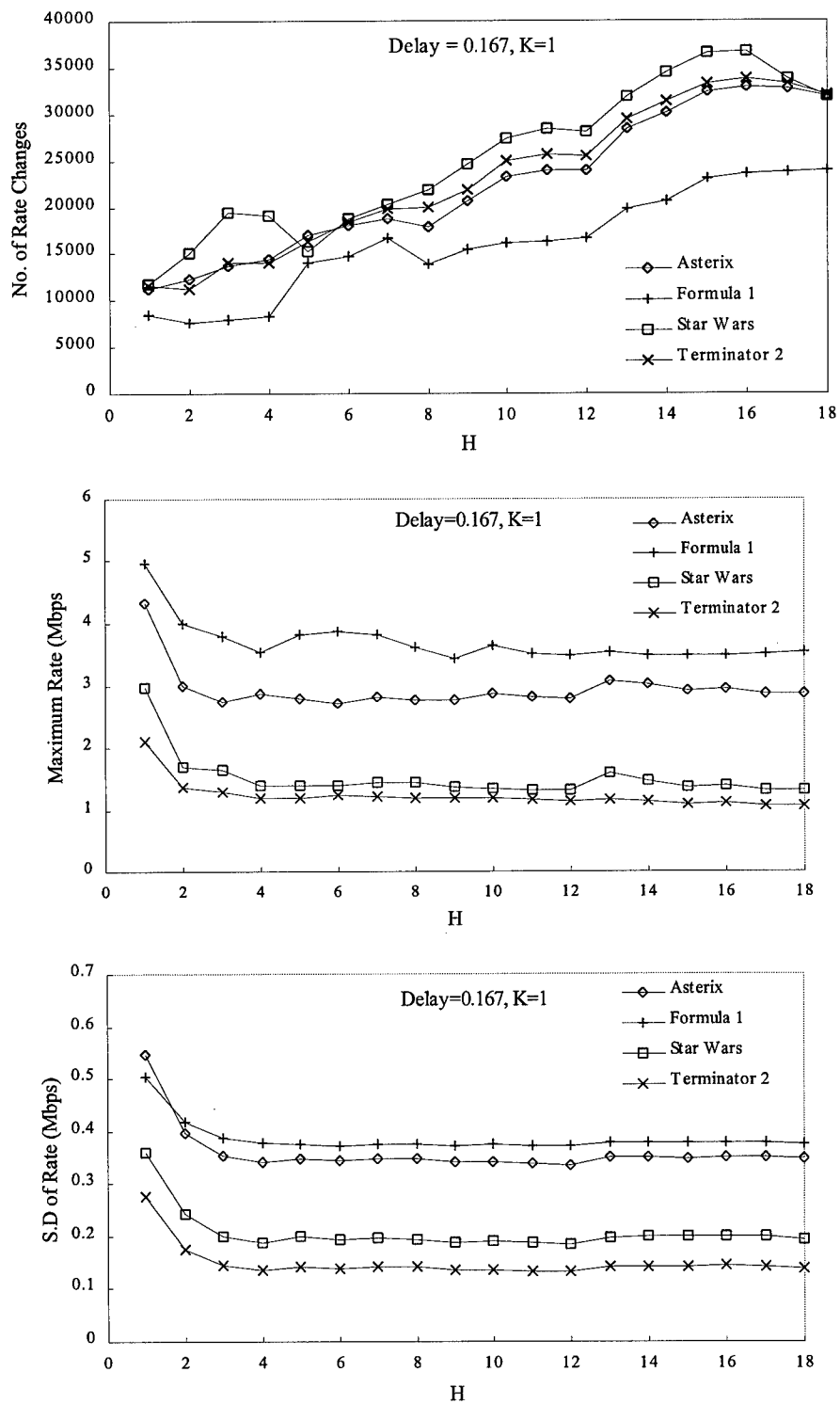


Figure 2.9: Performance of causal algorithm as a function of parameter H .

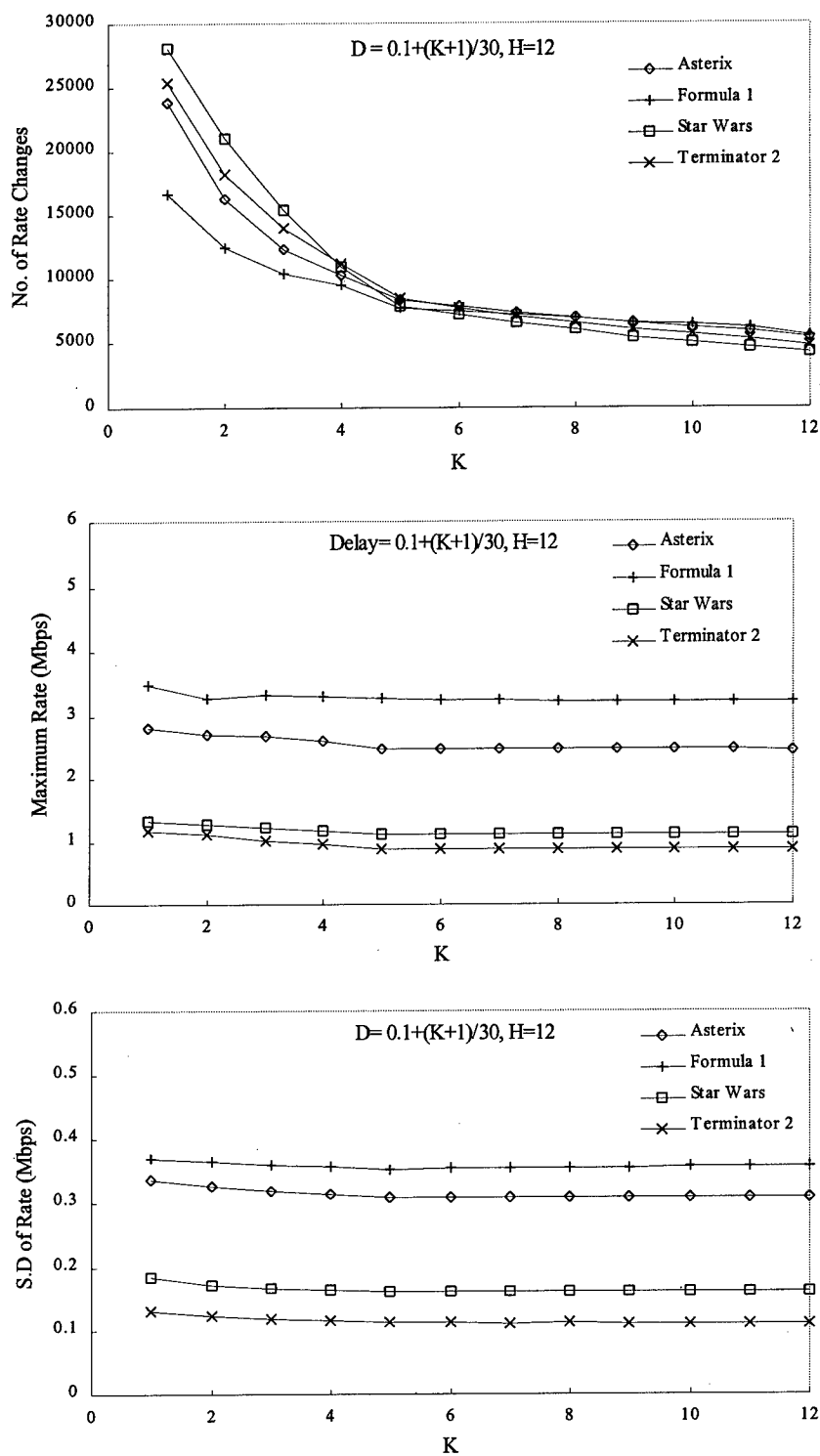


Figure 2.10: Performance of causal algorithm as a function of parameter K .

2.4.2 Improvement of Shortest Path Algorithm

The introduced algorithm gives relatively high number of rate changes due to the estimation error of the future picture sizes. From (2.3), it can be seen that lower bounds are derived using the size of pictures in the past and upper bounds are derived using the estimates of pictures in the future. Two observations can be made from the experiments in Section 2.4.1. First, lower bounds are less prone to the estimation error than the upper bounds due to higher number of known picture sizes. Second, choosing the lower path when destination point is reached, does not guarantee that the same rate can be chosen in the next period, instead rate should be chosen between the maximum and minimum values in order to increase the possibility that it falls within the new bounds in the next period. The effect of this choice would be an increase in the maximum rate and the standard deviation of rate. The following modifications to the specification of shortest path algorithm in Figure 2.4 decreases the number of rate changes significantly:

1. When a new cusp is found, no matter to which path it belongs, choose $\alpha=1$ (always choose the lower path or the minimum rate).

2. If destination point is directly connected to the source point, which corresponds to the case of rate being the same as in the previous period, keep it the same ($\alpha=1$), otherwise, take the average of maximum and minimum rates ($\alpha=0.5$).

The choice of $\alpha=0.5$ provides good performance in the average although an optimal value exists for each video sequence, but the experiments indicate that improvement

in the performance is not significant when the optimal value of alpha is used, so $\alpha=0.5$ is used for all cases.

2.4.3 Comparison of Smoothing Algorithms

The performance of the proposed algorithm can be judged better when it is compared with respect to other smoothing algorithms existing in the literature. Five of these algorithms address the problem of smoothing real-time VBR MPEG video with delay constraint. The first three algorithms have been described in [49] which are designed to minimize the maximum rate and the standard deviation of rate rather than number of rate changes, whereas the last two are similar except the last one improving on the previous one in the number of rate changes [13, 48].

Algorithm 1 (Uniform over delay interval) The data for each picture is spread out for transmission uniformly across D picture times, so that the data for that particular picture needs to be sent in one frame time is $E_i / (D - 1)$. This algorithm gives the worst performance since each picture is smoothed independently from the others not taking advantage of I-B-P pattern of MPEG video.

Algorithm 2 (Uniform over GOP interval) The objective is to attempt to achieve uniformity of the traffic over the entire GOP, to prevent bursts in the traffic profile related to occurrence of I or P frames. However, excessive delays occur if all frames within a GOP are buffered, so the algorithm finds the necessary number of bits that should be transmitted based on the buffer occupancy and delay constraint. This algorithm gives the best performance in the standard deviation of rate, but suffers from high number of rate changes.

Algorithm 3 (Transmit as fast as possible below peak rate) The basic idea is to transmit at the peak rate that has been determined to be needed, so long the bit rate is less than the peak and buffer is empty. This approach ensures that the buffer occupancy just before the start of the I-frame is as small as possible, so that the minimum rate for the I-frames is minimized. In fact, this algorithm gives the highest standard deviation since the transmission rate drops to the individual picture rates after the encoder buffer is empty. But, minimum peak rate is achieved compared to the first two algorithms.

Algorithm 4 The algorithm computes a lower bound for the transmission rate during each period such that the delay bound D is satisfied. Upper bounds for the rate are computed to ensure continuous transmission. If the rate during the previous period is between the upper and lower bounds of the current frame, the rate remains the same, otherwise, the rate is chosen as the upper or lower bound of the current frame which decreases the number of rate changes. The drawback of this algorithm is the high number of rate changes, which are small but frequent. This makes it difficult for network scheduling.

Algorithm 5 Another algorithm based on Algorithm 6 introduces two more parameters to choose the rate between upper and lower bounds instead of the exact bounds when a rate change is needed. In fact, the rate changes are decreased, in some cases, by a factor of four, but there are two main problems with this algorithm. First, it is difficult to predict the optimal values of those two parameters when encoding of the video is in real-time and significant increases in the maximum rate are observed especially for larger delays.

Algorithm 6 and 7 Original and improved causal algorithms respectively.

Table 2.2: Performance of 7 algorithms applied to Formula 1 and Star Wars MPEG Sequences.

| Algor. No | Formula 1 | | | Star Wars | | |
|--------------|------------------------|------------------------|------------------------|---------------------------|------------------------|----------------------|
| | No. of Rate Changes | Maximum Rate (Mbps) | S.D. of Rate (Mbps) | No. of Rate Changes | Maximum Rate (Mbps) | S.D of Rate(Mbps) |
| 1 | 39973 | 4.34304 | 0.38648 | 39903 | 1.42608 | 0.2 |
| 2 | 38793 | 3.34138 | 0.35546 | 30219 | 1.29996 | 0.17329 |
| 3 | 37440 | 3.33158 | 0.5046 | 23198 | 1.29966 | 0.19 |
| 4 | 12306 | 3.39466 | 0.38912 | 22046 | 1.30709 | 0.1916 |
| 5 | 5243 | 4.27505 | 0.394093 | 20934 | 1.547572 | 0.190818 |
| 6 | 13167 | 3.339203 | 0.361009 | 23589 | 1.29519 | 0.173892 |
| 7 | 4892 | 3.445754 | 0.368680 | 18304 | 1.395934 | 0.180995 |

Table 2.2 presents the results of applying seven algorithms to the video sequences of Formula 1 and Star Wars with $D=0.2$ seconds, $H=12$ and $K=1$. Algorithms 1-3 are implemented according to the descriptions given in [49] and Algorithms 4 and 5 using the specifications provided in [13] and [48] respectively. The optimal values of the two parameters needed by Algorithm 5 for Formula 1 and Star Wars sequences are used as provided in [48]. Algorithm 2 gives the smoothest rate in terms of standard deviation. Algorithm 7 provides the minimum number of rate changes and also very good performance in the standard deviation of rate whereas, Algorithms 2, 3, and 6 provide the smallest maximum rate. The results indicate that Algorithm 7 performs the best in overall with the minimum number of rate changes and standard deviation of rate at the cost of a minimal increase in the maximum rate.

The different behavior of the algorithms can be explained as follows. The first three algorithms utilize only the sizes of pictures in the buffer not taking advantage of future picture sizes resulting in relatively high number of rate changes, but also the smoothest

rate. Algorithms 4 and 5 utilize the size of pictures in the future to derive the upper and lower bounds by using simple estimates, which results in relatively worse performance compared to 6 and 7, since both upper and lower bounds are affected by the estimation error. On the other hand, the proposed algorithm uses the sizes of pictures in the past to derive the lower bounds and estimates of the pictures in the future to derive the upper bounds. This separation provides the flexibility to control the behavior of the algorithm, e.g., for better maximum rate and standard deviation, lower path is chosen (Algorithm 6), for minimum number of rate changes, the upper bounds are utilized (Algorithm 7).

In order to evaluate the performance of the algorithms as a function of delay bound, experiments using Algorithms 4-7 and non-causal algorithm which corresponds to the ideal smoothing were conducted. The results are given in Figures 2.11 and 2.12 for Formula 1 and Star Wars video sequences. As expected, Algorithm 7 is superior in the number of rate changes and the standard deviation of rate especially after $D=0.15$ seconds. The problems associated with Algorithm 5 can be seen by observing the significant increases in the peak rate and the standard deviation of rate for both video sequences. It should be noticed that Algorithm 6 allows for better tracking of input rate, thus better smoothing with smaller maximum rate and standard deviation, but with frequent changes of rate. For video bit streams with rapidly changing scenes and picture sizes, as in the case of Formula 1 trace, both Algorithm 6 and 7 are better choices if peak rate needs to be minimized when allocating network bandwidth. The effect of smoothing on Formula 1 is the drastic decrease in the number of rate changes but this does not apply to the case of Star Wars. This is because the difference between the upper and lower bounds is large

when input traffic is changing rapidly with large estimation error margin resulting in less number of rate changes. But for slowly changing traffic, rate prediction is heavily affected by the estimation error since error margin is smaller although when delay bound is allowed to be more than 0.2 second, the difference in the performance between ideal and causal algorithms is marginal.

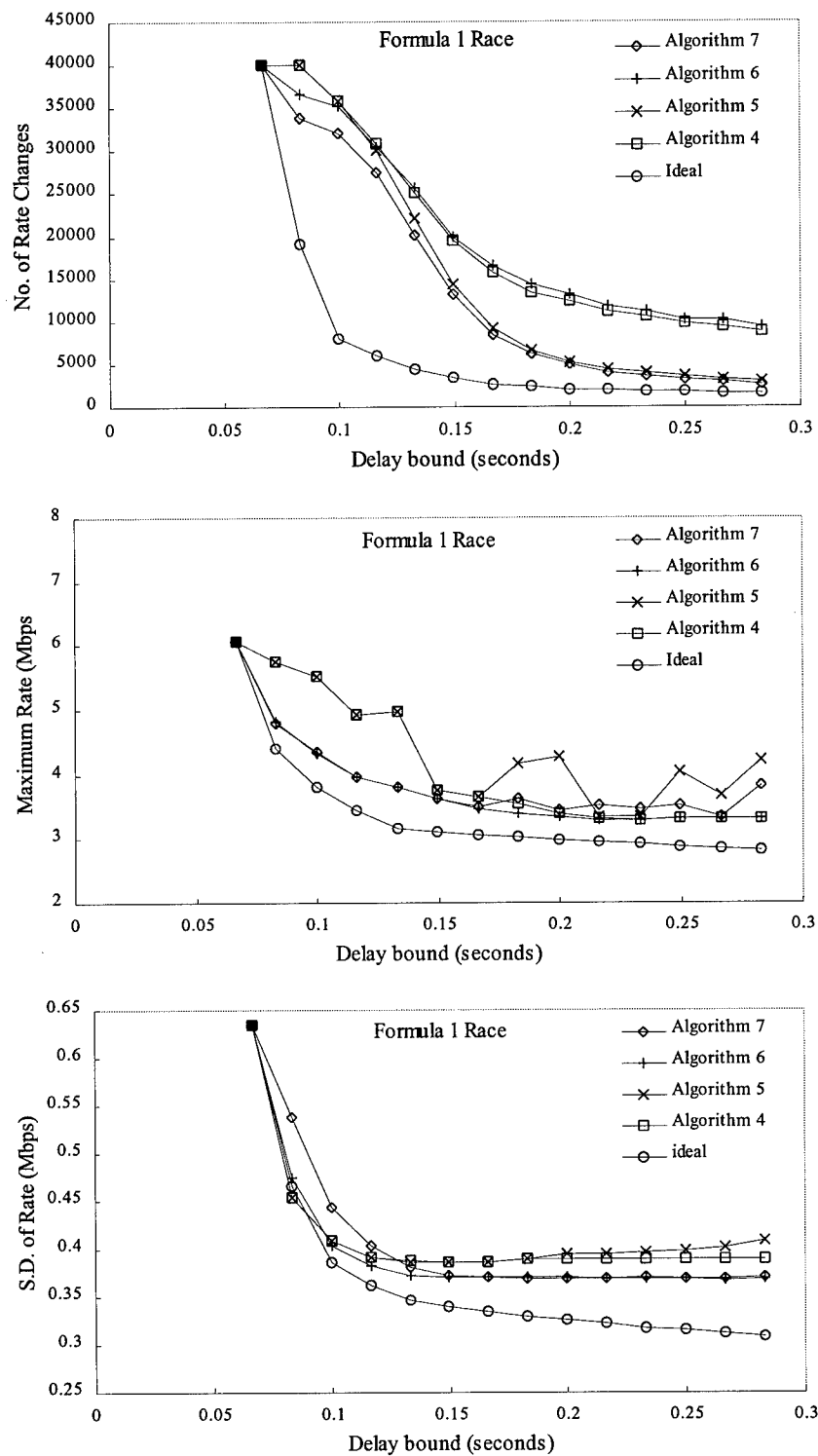


Figure 2.11: Performance of three algorithms as a function of delay bound for Formula 1.

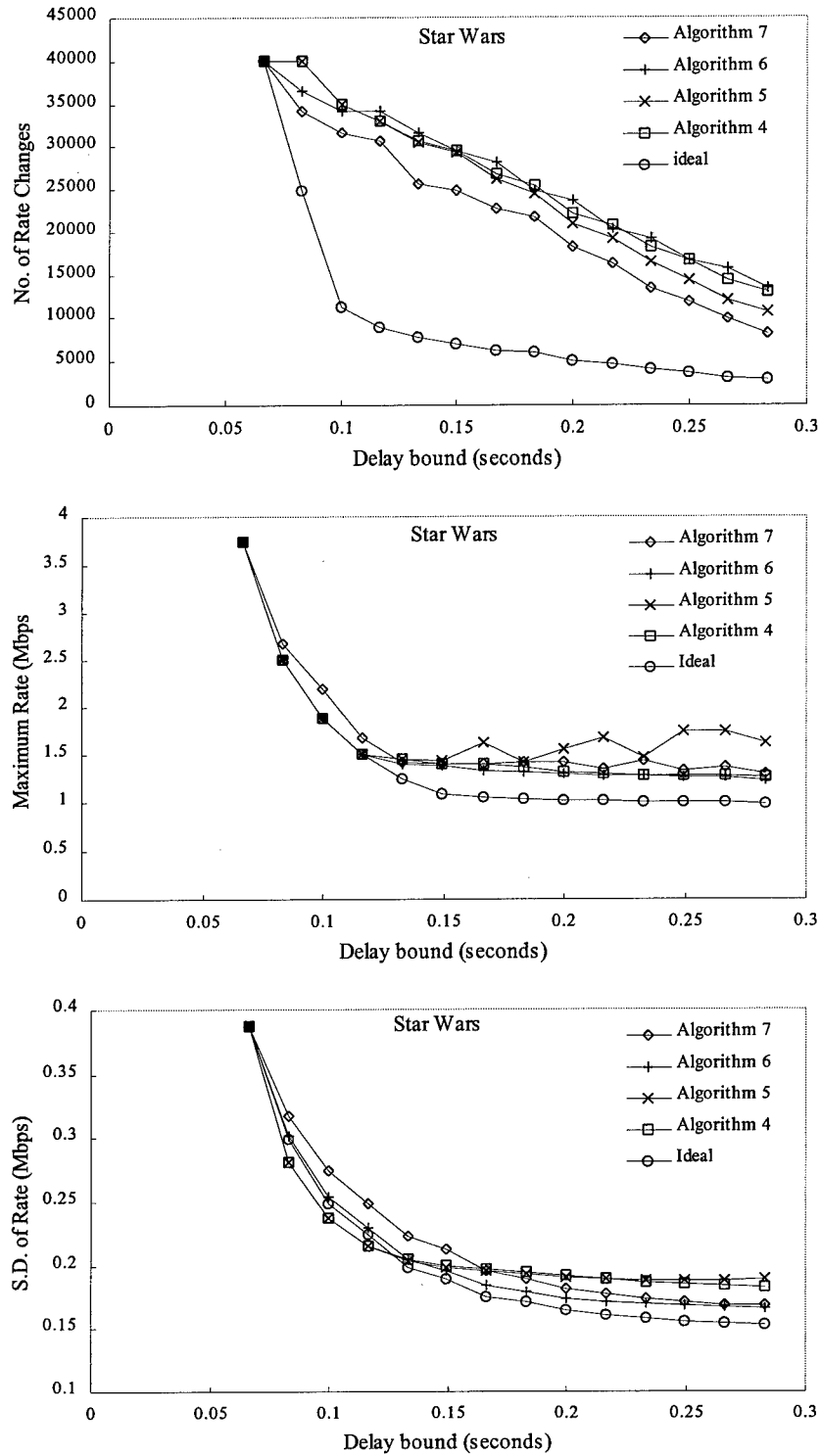


Figure 2.12: Performance of three algorithms as a function of delay bound for Star Wars.

2.4.4 Buffer vs Delay Constraint

The issue of using either buffer or delay constraint can be addressed when the availability of resources is a factor in the design of the system. Although either of the constraints provides effective smoothing, the performance of the smoothing algorithm for each constraint can be evaluated using the definition of a cost function as in the following:

$$\text{Total Cost} = \left(\max_i D_i \right) + \left(\frac{\max_i B_i}{E_{\text{avg}}} \right) \quad (2.8)$$

where D_i is the delay of picture i , B_i is the amount of data in the buffer at the end of period i , E_{avg} is the average picture size. The second term in (2.8) denotes the approximate number of pictures queued in the buffer.

A set of experiments was conducted to derive the cost of smoothing as a function of one of the three performance measures as defined in Section 2.4.2. The ideal smoothing algorithm was applied to frame traces of Formula 1 and Star Wars using buffering only at the client. Figures 2.13 and 2.14 show the results of the experiments for Formula 1 and Star Wars frame traces. Buffer constraint costs less when the number of rate changes is used as a performance measure except for very large delay bounds or buffer sizes where the cost is almost the same and delay constraint costs less for all possible delay bounds and buffer sizes when maximum rate is the performance measure. In the case of buffer constraint, the difference between lower and upper bounds is usually larger compared to the case of delay constraint resulting in longer intervals with constant rate, but also with

larger maximum rate since the delay constraint allows for better tracking of input rate. However, when standard deviation of rate is the performance measure, the statistics of the trace determines which constraint is more effective. In the case Formula 1 trace, delay constraint costs more, whereas for Star Wars trace, buffer constraint costs more. This is due to the fact that Formula 1 has many rapidly changing scenes resulting in large increases or decreases in picture sizes even for the same picture type (I, B, or P) which can be smoothed more effectively when the difference between upper and lower bounds is sufficiently large or almost constant as in the case of buffer constraint. Star Wars trace includes slowly changing scenes and is smoother (size of pictures of the same type does not change rapidly in the consecutive scene) compared to Formula 1 trace so delay constraint is more effective in tracking the input traffic.

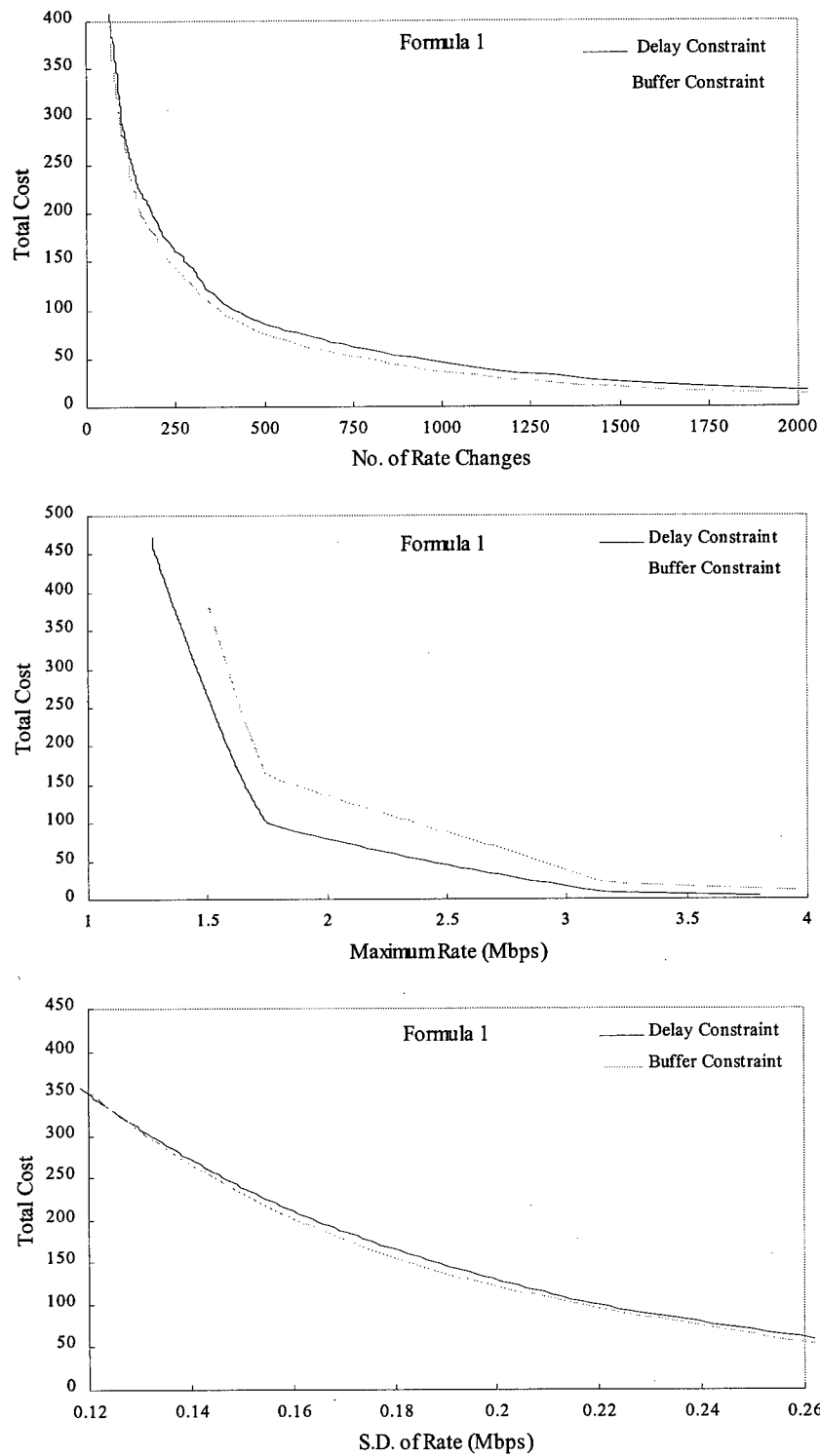


Figure 2.13: The total cost as function of three performance measures for Formula 1.

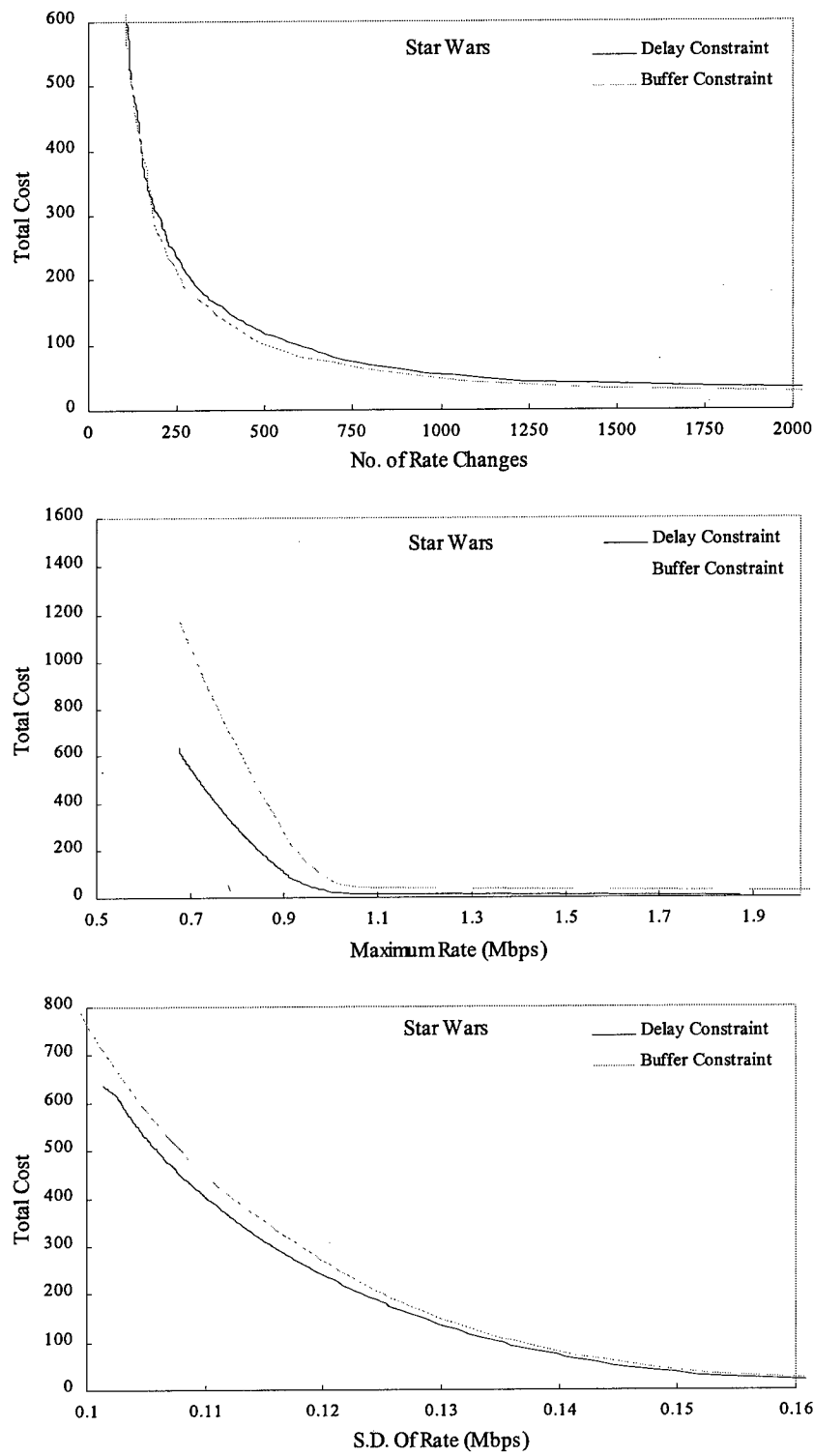


Figure 2.14: The total cost as function of three performance measures for Star Wars

2.5 Conclusion

The smoothing of video traffic plays an important role in the design of video communication systems. As part of a research project on the design of transport and network protocols for multimedia applications, a smoothing algorithm is specified and designed to satisfy a given set of buffer and delay constraints. A large set of experiments was conducted using statistics of MPEG video sequences to study the performance of the algorithm. It has been shown that the algorithm is effective in smoothing VBR video when its performance is compared with respect to other techniques existing in the literature. The design of the algorithm allows the users to choose the optimal behavior for a given performance measure including the number of rate changes, the maximum rate and the standard deviation of rate. This feature will be exploited in the next chapter when renegotiating bandwidth with the network.

The proposed algorithm can be used as part of a video transport system where the QoS of the underlying network services may change during the transmission. For networks with no QoS guarantees, the algorithm can be extended to include varying network conditions as well. Since the foundation of the model is based on the status of buffers at the server and client, the algorithm can adapt to changing network conditions by using the feedback from the client to recompute the bounds based on information about the buffer occupancies at the client and server.

Chapter 3

Effects of Smoothing on End-to-end Deterministic Guarantees for VBR Traffic

3.1 Introduction

Future packet-switching integrated services networks must provide support for distributed multimedia applications with stringent delay and loss requirements in terms of network performance. Of the many traffic classes in integrated services networks, delay and loss sensitive VBR video traffic will be generated by most distributed multimedia applications. For bursty VBR traffic, it is generally difficult to provide the good QoS that the network clients specify for, and to simultaneously achieve high network utilization. There has been many research on the issue of providing services with different degree of QoS guarantees including deterministic, statistical and best-effort services. Among these, deterministic service guarantees that all packets of a connection will meet the promised QoS, whereas with statistical service, only probabilistic performance bounds are guaranteed. These two services can be viewed as a tradeoff between the QoS of the connection and higher network utilization; since statistical multiplexing results in overallocating of network resources. However, deterministic performance guarantees are especially impor-

tant for time-critical and loss-sensitive data given recent studies on the effect of cell loss on the perceptual quality of video [68]. Contrary to conventional wisdom, deterministic service does not require a peak-rate-allocation scheme and reasonable network utilization can be achieved even while providing worst-case guarantees through better traffic models such as Deterministic Bounding Interval Dependent (D-BIND) traffic model [69], and with more accurate admission control schemes as in [70-73].

An important issue in providing end-to-end performance guarantees is the effect of smoothing bursty traffic on network utilization and QoS. In Chapter 3, the effect of smoothing on the specification of UPC for ATM networks is investigated and it is found that with smoothing of bursty traffic, the cost of transmission can be reduced by allocating less amount of network resources than the case for unsmoothed traffic. While reducing the burstiness of VBR sources through traffic smoothing may help users reduce their cost and at the same time increase network utilization, it also introduces an increase in the end-to-end delay by contributing extra delay at the network client buffer. In this thesis, the type of smoothing in which bursts are spread over time by adding variable delay to packets is considered instead of reducing source's bandwidth or dropping packets during bursts both of which deteriorate the perceptual quality of the video [33, 68]. The smoothing scheme proposed in Chapter 2 is used which provides a bounded delay at the buffer for each packet sent to the network.

The input traffic is characterized by D-BIND traffic model using bounding rates over multiple interval lengths which allows for a higher network utilization by providing a more accurate traffic specification [69]. With the D-BIND traffic model, it is possible to

define the relative burstiness of input traffic in a manner similar to [74]. When the smoothed, less bursty traffic sources are multiplexed at queues inside the network, the resulting bound on queuing delay will be reduced. This allows for more admissible connections or better QoS support given the same number of connections. However, the extra smoothing delay must be accounted for when considering the total end-to-end delay bound. Thus, in this chapter, the effect of smoothing VBR video on the end-to-end delay bound which consists of smoothing delay at the client source buffer and queuing delay inside the network is investigated. Since smoothing decreases the bound on queuing delay, but increases the bound on smoothing delay, it can be considered as a tradeoff between buffering at the source and buffering inside the network. For the case of ideal smoothing where future traffic is known, it is shown both analytically and empirically that the extra delay contributed by the smoothing of a source is equal to the gain in queuing delay when multiplexing smoothed sources over a congested hop with homogeneous sources resulting in non-negative savings in the end-to-end delay bound. In a similar work, smoothing over a single hop has been found ineffective due to the traffic shaping implemented by a FIFO which services packets at a smoothing rate R_s where R_s is less than the unsmoothed source's peak rate and greater than its long term average rate [43]. In contrast, results presented in this thesis indicate that, with ideal smoothing, it is possible to achieve higher network utilization without any degradation in the QoS of the connection even for a single hop. Alternatively, for multiple congested nodes, smoothing results in significant reductions in the end-to-end delay bound since sum of the savings in queuing delay at each congested hop is more than the incurred extra smoothing delay at

the source. Thus, a higher network utilization or lower end-to-end delay can often be achieved by smoothing. This result also applies to real-time traffic which can not be as efficiently smoothed as stored video. The experiments indicate that smoothing of real-time traffic is beneficial only when there exists a minimum number of congested hops in the path of the connection and when traffic is smoothed over at least four frame periods as smoothing has been shown to be most effective for delays over three frame periods in Chapter 2 and 3.

In the recent literature, traffic shaping or smoothing has received much attention. For example, in [43], authors use similar techniques to those used here to determine the conditions for which smoothing will result in a net reduction in end-to-end delay bound when traffic shaping is realized by a FIFO buffer with constant service rate. In another similar work, the authors show that end-to-end delays with rate controlled services are strictly less than with Rate Proportional Processor Sharing [45].

In a related work on deterministic guarantees, a traffic shaper which affects peak rate or cell spacing has been considered and for a single hop, it was argued that significant utilization gains are possible [46], however this gain is overstated since a peak-rate-allocation is assumed which was shown to be an unnecessary condition for providing deterministic service [71, 73]. Several other work consider the case of traffic shaping in terms of statistical performance guarantees as in [47]. Another work considers deterministic smoothing of MPEG traffic sources [75] then provides statistical guarantees using histogram techniques introduced in [76]. In this chapter, the deterministic approach that does not allow the traffic shaper to discard packets is considered.

This chapter is organized as follows. In Sections 3.2 and 3.3, the D-BIND traffic model and its associated admission control test are reviewed for a First Come First Serve (FCFS) scheduler with no-loss, no-delay-violation deterministic guarantees. In Section 3.4, it is proven analytically that for a single congested hop, when ideal smoothing is applied to VBR sources, the reduction in queuing delay bound is equal to the additional smoothing delay when all sources have the same smoothing delay bound. In Section 3.5, the experimental results are presented for the traces of MPEG video to show the effectiveness of smoothing and tradeoffs for both stored and real-time video. Finally, Section 3.6 gives some concluding remarks.

3.2 The Deterministic D-BIND Model

Compared to statistical service, deterministic service provides better QoS in the sense that it provides no-loss no-delay service. For the network to provide such service, a deterministic upper bound is required on all sources receiving this service. This allows enforcement of source's traffic specification by using schemes like leaky-bucket policer. On the other hand, statistical models of the source are much more difficult to enforce.

The Deterministic Bounding Interval Dependent (D-BIND) has been introduced in [69] to capture the property that sources exhibit burstiness over a wide variety of interval lengths. The key components of D-BIND model are that it is *bounding*, required to provide deterministic QoS guarantees, and *interval-dependent*, needed to capture the burstiness properties of sources. With respect to other models introduced in [77], this more ac-

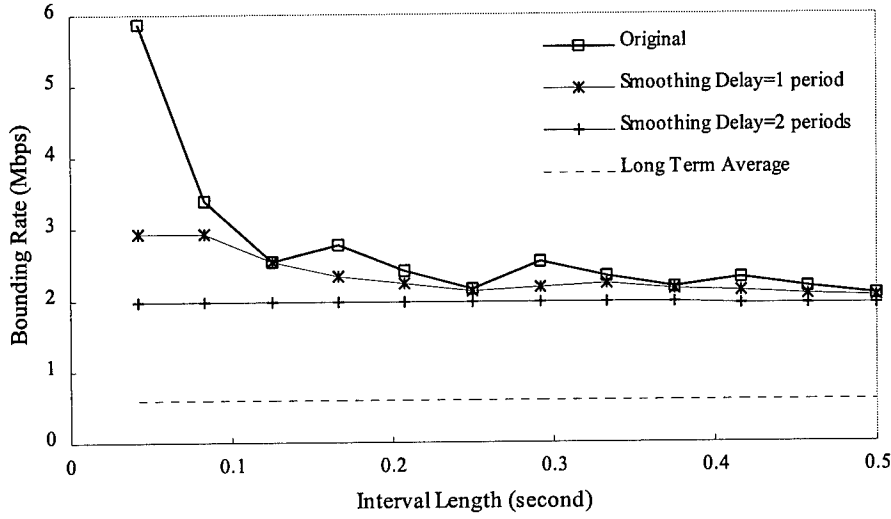


Figure 3.1: D-BIND rate-interval pairs for a segment of Goldeneye Movie.

curate traffic characterization allows for a higher network utilization for a given delay bound.

D-BIND traffic model defines a traffic constraint function $b(t)$ which constraints or bounds the source over every interval of length t . Denoting $A[t_1, t_2]$ a connection's arrivals in the interval $[t_1, t_2]$, $b(t)$ requires that $A[s, s+t] \leq b(t)$, $\forall s, t > 0$. The D-BIND model is defined via rate-interval pairs $\{(R_k, I_k) | k = 1, 2, \dots, P\}$. The constraint function is then defined as a piece-wise linear function

$$b(t) = \frac{R_k I_k - R_{k-1} I_{k-1}}{I_k - I_{k-1}} (t - I_{k-1}) + R_{k-1} I_{k-1}, \quad I_{k-1} \leq t \leq I_k \quad (3.1)$$

with $b(0) = 0$. Thus the rates R_k are an upper bound on the rate over every interval of length I_k so that

$$\frac{A[t, t + I_k]}{I_k} \leq R_k \quad \forall t > 0, k = 1, 2, \dots, P. \quad (3.2)$$

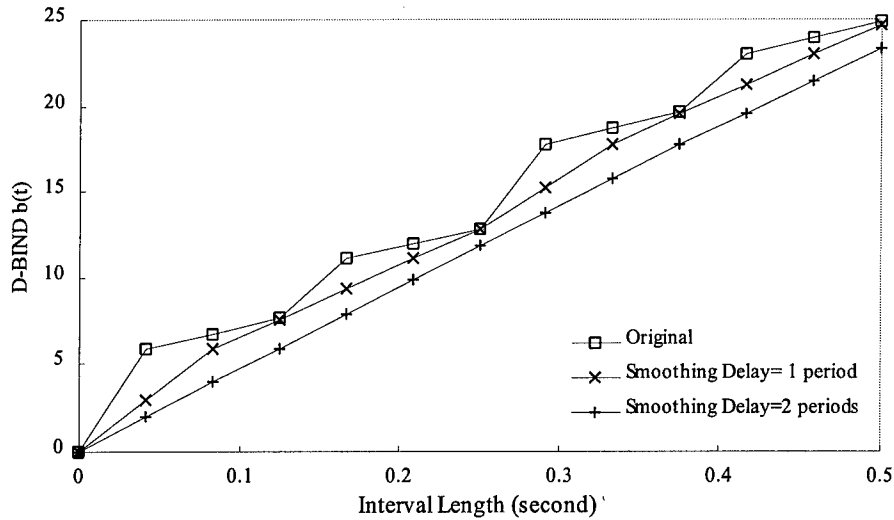


Figure 3.2: D-BIND constraint function for Goldeneye sequence.

In Figure 3.1, a plot of the D-BIND rate-interval pairs for a 40,000 frame trace of MPEG compressed James Bond: Goldeneye movie is shown. It can be observed that, for small interval lengths, R_k approaches the source's peak rate while for longer interval lengths, it approaches the long term average rate for the original traffic. The effect of smoothing is the reduction in R_k for small interval lengths leading to almost constant value of R_k for all interval lengths when smoothing delay is increased.

Figure 3.2 shows the D-BIND constraint function $b(t)$ described by (3.1) for the same movie. It can be seen that the D-BIND model captures the temporal properties of the MPEG video. For example, the peak rate of the original unsmoothed traffic is caused by the largest I-frame of the sequence giving the initial slope. Next, the slope decreases with the transmission of a P-frame which is usually smaller than an I-frame. The effect of smoothing on the D-BIND constraint function is the smaller amount of traffic within an interval and slope becoming constant with increasing smoothing delay.

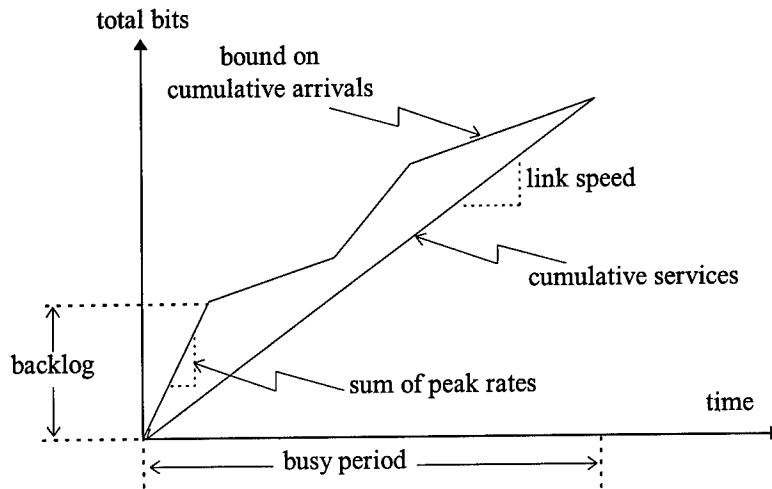


Figure 3.3: Deterministic Admission Control.

3.3 Connection Admission Control

Deterministic admission controls rely on the delay analysis techniques of [71, 73] which are illustrated in Figure 3.3. The upper curve represents the total number of cumulative bits that have arrived from all sources into the queue by time t and the lower curve represents the total number of bits transmitted by time t . The difference between the two curves is the number of bits currently in the queue, or the backlog function. When the backlog function is zero, there are no bits in the queue and thus a busy period has ended. If the upper curve is a deterministic bounding curve then the maximum delay can be expressed as a function of two curves. For example, the maximum backlog divided by the link speed provides an upper bound on delay for a FCFS scheduler [72].

The constraint function provides the required bound on arrivals in any interval of length t , so that with the aggregate of individual source's respective $b(t)$ constraint functions forming the upper curve of Figure 3.3, admission control conditions for determinis-

tic delay and throughput may be derived. For example, for a FCFS scheduler with $j=1, 2, \dots, n$ multiplexed connections constrained by their respective constraints $b_j(t)$, and with a link speed l , and a maximum packet size \bar{s} , a deterministic upper bound on delay for all connections is given by

$$d = \frac{1}{l} \max_{t \geq 0} \left\{ \sum_{j=1}^n b_j(t) - lt + \bar{s} \right\} \quad (3.3)$$

The proof is given in Theorem 1 of [73]. In the homogenous case, the maximum number of connections that can be multiplexed for a delay bound d is therefore given by

$$N(d) = \max \left\{ n \mid \frac{1}{l} \max_{t \geq 0} \{nb(t) - lt + \bar{s}\} \leq d \right\} \quad (3.4)$$

3.4 Effect of Smoothing on the Deterministic Service

Smoothing of a VBR traffic from D-BIND model point of view is the transformation of a source with upper bounds $\{(R_k, I_k) \mid k = 1, 2, \dots, P\}$ to $\{(\hat{R}_k, \hat{I}_k) \mid k = 1, 2, \dots, P\}$, with $\hat{R}_k \leq R_k$ if $\hat{I}_k = I_k$ as it can be seen in Figure 3.1. A second view is the transformation of source's D-BIND constraint function $b(t)$ to a new constraint function $\hat{b}(t)$:

$$\hat{b}(t) = \frac{\hat{R}_k \hat{I}_k - \hat{R}_{k-1} \hat{I}_{k-1}}{\hat{I}_k - \hat{I}_{k-1}} (t - \hat{I}_k) + \hat{R}_k \hat{I}_k, \quad \hat{I}_{k-1} \leq t \leq \hat{I}_k \quad (3.5)$$

with $\hat{b}(t) \leq b(t) \forall t$ from Equations (3.1) and (3.5) when $\hat{R}_k \leq R_k$. In this chapter, the formal definition of smoother traffic is used in the following manner similar to those in [43, 74]:

Definition 3.1 If $\lim_{t \rightarrow \infty} \frac{b(t)}{t} = \lim_{t \rightarrow \infty} \frac{\hat{b}(t)}{t}$, then $\hat{A}(t)$ is considered smoother or less bursty than $A(t)$ if $\hat{b}(t) \leq b(t) \forall t$.

An example of the transformation of smoothing on a source's constraint function $b(t)$ can be seen in Figure 3.2 for two smoothing delay values of 1 and 2 frame periods.

From network utilization point of view, less bursty sources lead to better network utilization which is stated in the following Lemma [43]:

Lemma 3.1 If a source j is smoothed so that the arrival process $\hat{A}(t)$ is less bursty than $A(t)$, then the queuing delay bound for a FCFS scheduler is reduced.

The proof of Lemma 3.1 is obvious by using Equations (3.1) and (3.3) and the fact that $\hat{b}_j(t) \leq b_j(t) \forall t$. However, while smoothing reduces the queuing delay bound, it also introduces an additional delay due to buffering at the client. The worst case smoothing delay has been defined in [71] as the maximum horizontal time distance between the two curves $b(t)$ and $\hat{b}(t)$. That is,

$$\Delta_s = \max_{t_2 > t_1} \{t_2 - t_1 \mid \hat{b}(t_2) = b(t_1)\} \quad (3.6)$$

The worst case smoothing delay contributes to the total end-to-end delay perceived by the source. A source should be smoothed if the additional delay bound is less than the reduction in the queuing delay bound. Below, the conditions under which a source should be smoothed are derived when ideal smoothing function is used over a single hop. But first, the following notations are defined: d represents the queuing delay bound for an

unsmoothed source, \hat{d} represents the queuing delay bound for the smoothed source and Δ_s represents the worst case smoothing delay. The following Lemma will be used in the proof of Theorem 3.1.

Lemma 3.2 If ideal smoothing function with maximum smoothing delay of Δ_s is applied to $A(t)$, then $A(t) \leq \hat{A}(t + \Delta_s) \forall t$ or equivalently $b(t) \leq \hat{b}(t + \Delta_s) \forall t$.

Proof: When ideal smoothing function is used with only delay constraint at the client buffer, the upper bound represents the original input traffic, whereas the lower bound represents input traffic with a time-shift of Δ_s corresponding to the worst case smoothing delay. Since, the ideal path crosses between the upper and lower bounds, in the worst case, $A(t) = \hat{A}(t + \Delta_s)$, corresponding to the case when the shortest path passes through a lower bound. Then $A(t) \leq \hat{A}(t + \Delta_s) \forall t$ for the general case.

The ideal smoothing function uses the delay to its possible extent, since it searches for the shortest path through the bounds. This implies that when a large burst arrives at time t_k , the smoothing function guarantees that $A(t_k) = \hat{A}(t_k + \Delta_s)$ since the minimum slope of the shortest path is obtained by crossing through the lower bound rather than the upper bound which would mean a rapid increase in the slope.

3.4.1 The Single Hop Case

The following theorem shows that for a single-hop network and homogenous sources, ideal smoothing does result in zero increase in end-to-end delay bound.

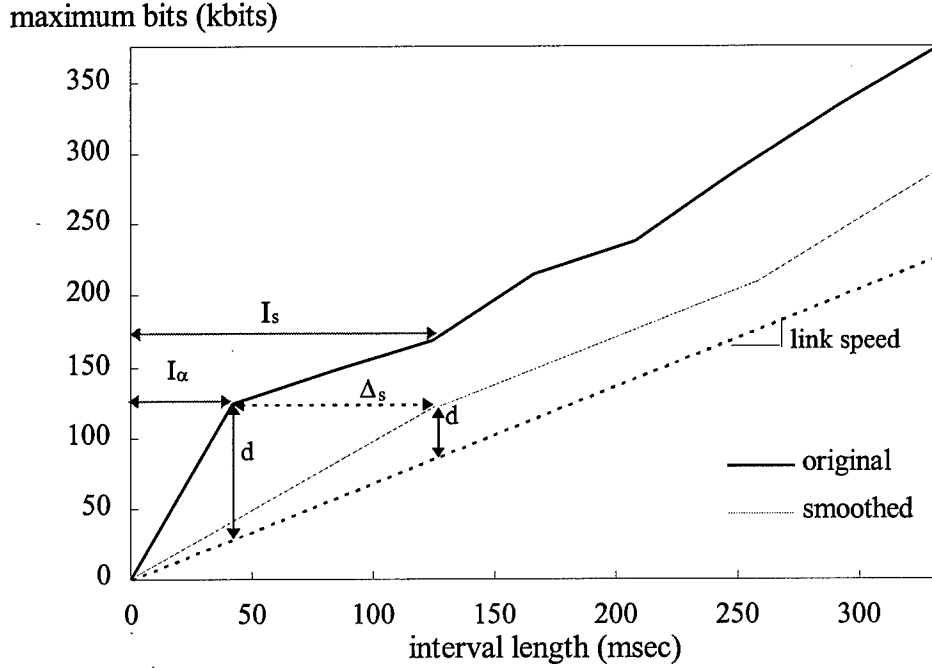


Figure 3.4: Effect of smoothing on network queuing delay bound.

Theorem 3.1 In the single-hop case for homogenous sources and deterministic delay bounds, the reduction in the queuing delay bound, $d - \hat{d}$, introduced by ideal smoothing function is equal to the worst case smoothing delay, Δ_s , when $d \geq \Delta_s$.

Proof:

For N homogenous sources described by the D-BIND traffic model parameters $(R_k, I_k)_{k=1}^P$, and a FCFS scheduler with link speed l , the queuing delay bound for all unsmoothed sources can be obtained using Equation (3.3):

$$d = \frac{1}{l} \max_{1 \leq j \leq P} \{ N b(I_j) - l I_j \}.$$

If this delay bound occurs at the interval length I_α , then $d = \frac{(N R_\alpha I_\alpha - l I_\alpha)}{l}$ is ob-

tained. Let the delay bound for smoothed sources be $\hat{d} = \frac{(N \hat{R}_s \hat{I}_s - l \hat{I}_s)}{l}$ as shown in

Figure 3.4. Using Lemma 3.2, $b(I_\alpha) \leq \hat{b}(I_\alpha + \Delta_s)$. Since all bits in the interval I_α are sent in the interval $I_\alpha + \Delta_s$ for the worst case, $b(I_\alpha) = \hat{b}(I_\alpha + \Delta_s) = \hat{b}(\hat{I}_s)$ is obtained. This corresponds to $b(I_\alpha) = R_\alpha I_\alpha = \hat{b}(\hat{I}_s) = \hat{R}_s \hat{I}_s$. By substituting $R_\alpha I_\alpha$ with $\hat{R}_s \hat{I}_s$ and \hat{I}_s with $I_\alpha + \Delta_s$ in Equation (3.3) for \hat{d} , $\hat{d} = d - \Delta_s$ is obtained when $d \geq \Delta_s$ since \hat{d} can not be negative. So this completes the proof.

An immediate implication of Theorem 3.1 is that when ideal smoothing function is used, higher network utilization can be obtained over a single congested hop without affecting end-to-end delay bound of the connection. This result is particularly important since it shows there exists a smoothing function which can be used to increase network utilization without any degradation in the QoS provided to the smoothed sources. Using other smoothing schemes has been found to be ineffective when there exists only a single congested hop along the path [43, 44].

3.4.2 The Multi-Hop Case

Theorem 3.1 showed that over a single hop, ideal smoothing function is not an effective means for achieving better QoS since net saving in its end-to-end delay bound is zero. However, over multiple hops, queuing delays may be incurred at more than one node, while the smoothing delay is incurred only once at the source's traffic shaper. Thus, a smoother source can reduce its queuing delay at each congested hop resulting in a net saving in its end-to-end delay bound. The effectiveness of smoothing will depend on the network load, the number of hops traversed, the burstiness of the stream, and the desired

delay bound. The following proposition is given in [43] which provides a rule for determining if smoothing provides a net advantage in terms of end-to-end delay bound for networks that use rate controlled service disciplines such as Rate Controlled Static Priority, Earliest Deadline First, or Hierarchical Round Robin [78]. Rate-controlled service schemes reshape the traffic at each hop by using leaky buckets at each node rather than only at the entrance of the network.

Proposition 3.1 If \hat{d}_i is the queuing delay bound at hop i for the smoothed source and d_i is the original queuing delay bound at hop i , a source will obtain a net reduction in end-to-end delay bound due to smoothing if the following condition holds:

$$\Delta_s \leq \sum_{i=1}^H (d_i - \hat{d}_i) \quad (3.7)$$

where H is the number of hops between the source and destination for networks using a rate-controlled service discipline.

The proof of Proposition 3.1 is given in [43]. Theorem 3.2 provides the minimum number of congested hops in order to obtain a net reduction in end-to-end delay bound when ideal smoothing function is used.

Theorem 3.2 In the multiple-hop network with homogeneous sources and deterministic delay bounds, a source will obtain a net reduction in end-to-end delay bound due to ideal smoothing if the number of congested hops in the path is at least two.

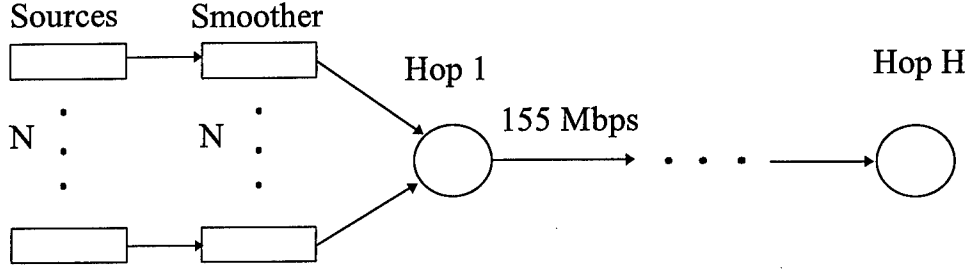


Figure 3.5: Network topology used in the experiments.

Proof: The net reduction in end-to-end delay bound is zero for the single-hop case so

$\Delta_s = d_1 - \hat{d}_1$. If there is more congested node in the path, then it is obvious that

$\Delta_s < \sum_{i=1}^2 (d_i - \hat{d}_i)$ since $d_2 - \hat{d}_2 > 0$. So this completes the proof.

3.5 Experimental Results

A segment of Formula 1 Race MPEG compressed video is used in order to verify the results of Theorem 3.1 and 3.2. The trace includes 10,000 frames corresponding to a total running time of 7 minutes at a 24 frames/sec display rate. Only network services with deterministic QoS guarantees are considered where the network provides a no-loss, no-delay violation guarantees. The experiments consider both the single and the multiple hop cases with the network topology shown in Figure 3.5. N connections are smoothed and then they are multiplexed at the network nodes. The packets of a connection traverse H hops until they reach their destination.

The experiments are conducted as follows. First the MPEG trace is used to calculate the source's D-BIND parameters under the various smoothing delays each of which expressed by the parameter S , the number of frames smoothed over. Then, the admission control test of Equation (3.3) is applied to determine the maximum number of homoge-

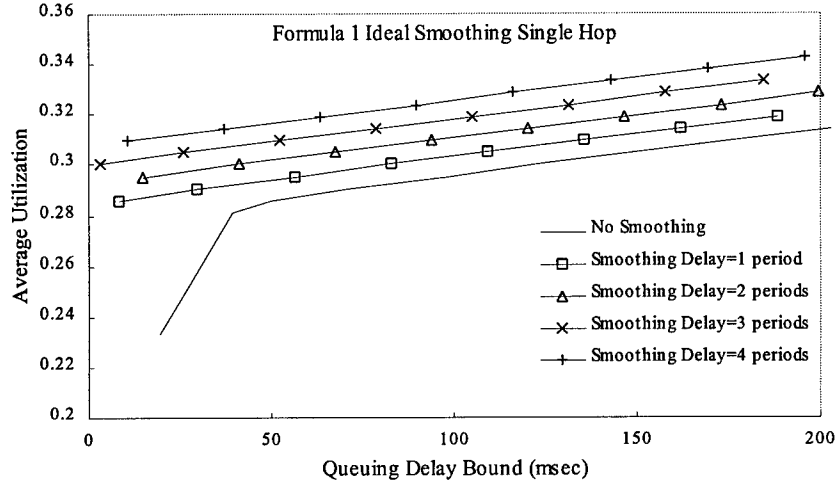
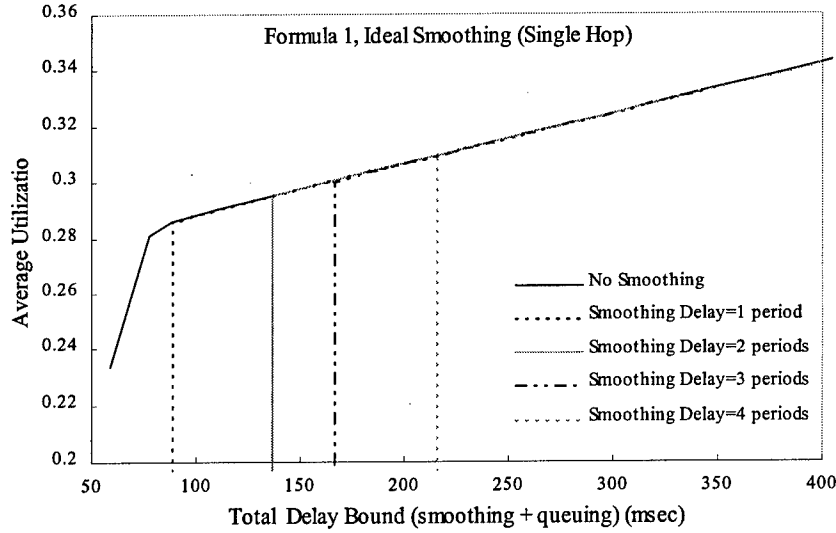


Figure 3.6: Average utilization of the multiplexer for various smoothing delays.

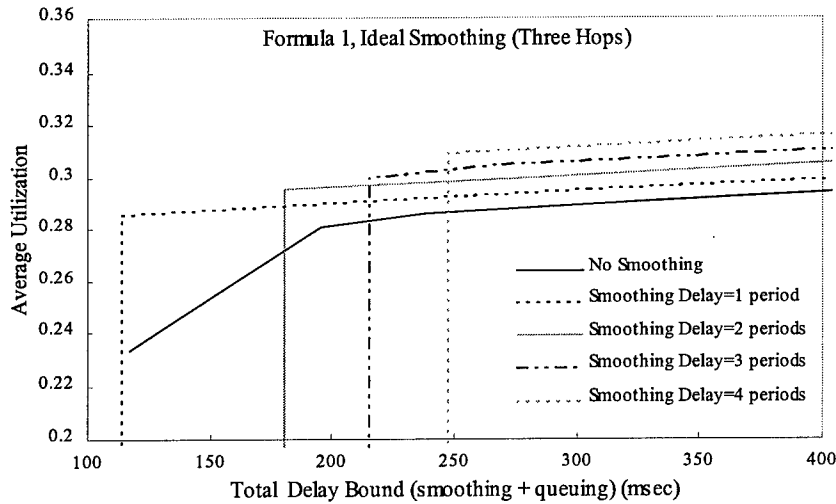
nous admissible connections on the 155 Mbps link for a given queuing delay bound. The following performance measures are used to evaluate the effectiveness of the smoothing function: average utilization of the network which is calculated as $N \cdot M / 155 \cdot 10^6$ where M is the long term average bit rate of the source in bits/sec; the net end-to-end savings in delay bound from smoothing connection j , $D_j - \hat{D}_j = \Delta_{s,j} + \sum_{i=1}^H (d_{i,j} - \hat{d}_{i,j})$ where D_j and \hat{D}_j correspond to the total end-to-end delay bound for original and smoothed source j ; and the total end-to-end delay bound $\hat{D}_j = \sum_{i=1}^H \hat{d}_{i,j} + \Delta_{s,j}$.

3.5.1 Ideal Smoothing (Stored Video)

In the first set of experiments, the average utilization of the multiplexer is used as the performance index. Figure 3.6 shows the effect of smoothing on the queuing delay bound, that is the delay experienced at a network node. The general shape of the curves indicates that as the queuing delay bound increases, more connections are admissible so that a higher utilization is possible. As stated in Lemmas 3.1 and 3.2, when the smoothing



(a)



(b)

Figure 3.7: Average utilization as a function of total end-to-end delay bound (a) for the single hop and (b) for three hops.

delay increases (from 0 to 4 in this case), the traffic transmitted to the network becomes smoother so that the queuing delay bound is reduced or equivalently, for a given queuing delay bound, higher utilization is achievable in the network.

When the total end-to-end delay bound is considered including both smoothing and queuing delays, the average utilization is the same for all smoothing delays for the single-

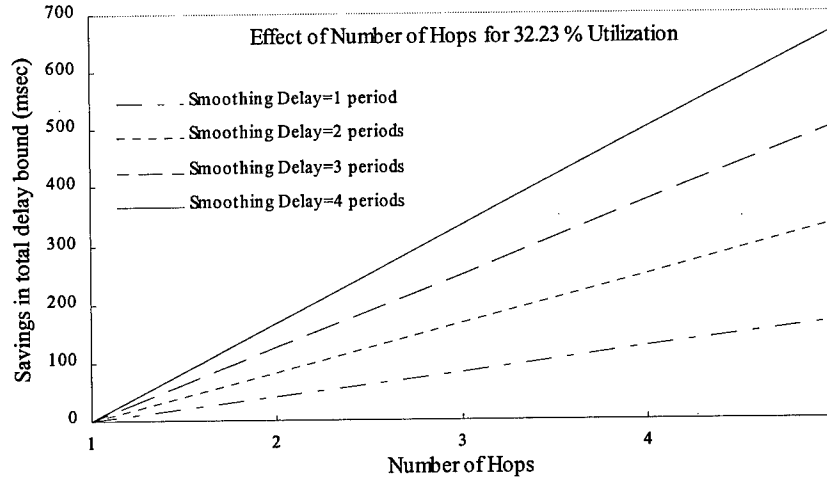


Figure 3.8: Effect of the number of hops on the savings in total delay bound.

hop case as shown in Figure 3.7(a). As stated in Theorem 3.1, there is no increase in the total end-to-end delay bound for the single hop case when the queuing delay is larger than smoothing delay of the source. Figure 3.7(b) shows the same experiment except that the sources traverse three hops rather than one. In this case, smoothing results in a substantial reduction in a source's end-to-end delay bound which was explained by Theorem 3.2: if there is more than one congested node in the path, end-to-end delay bound is reduced. Or equivalently, for a given end-to-end delay bound, more connections can be admitted to the network.

Figure 3.8 shows the effect of the number of hops on the savings in end-to-end delay bound for a fixed number of connections so that the average network utilization is 32.23% in all cases. In the homogenous case, $D - \hat{D}$ reduces to $H(d_j - \hat{d}_j) - \Delta_s$ so the savings in delay bound increases linearly with the number of hops. As predicted by Theorem 3.2, the lines start at zero for a single hop, then become positive with the two hops. With larger smoothing delays, more gain is obtained as expected.

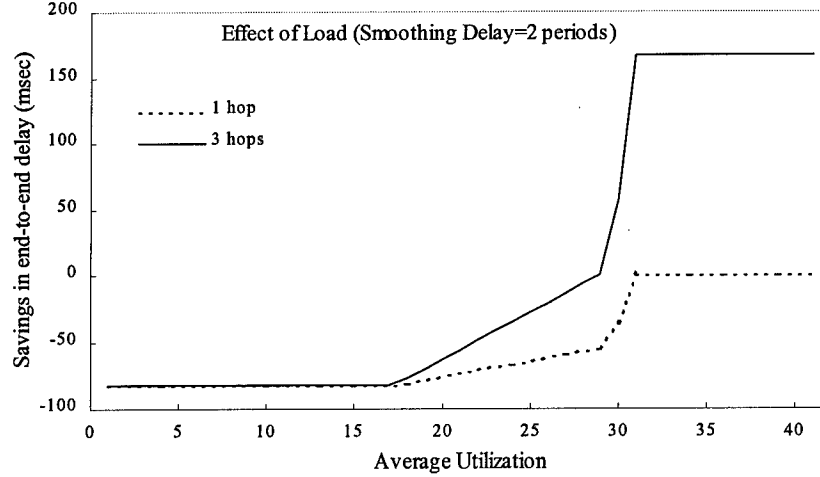


Figure 3.9: Effect of network load.

Figure 3.9 shows the savings in end-to-end delay bound as a function of network utilization. The smoothing delay is fixed to $S = 2$. As it can be observed, when the network is lightly loaded, smoothing does result in a negative delay saving for both one-hop and three-hops cases. From Theorem 3.1, it is known that there is a non-negative saving in total delay only if the smoothing delay is smaller than the queuing delay bound at a given network utilization for unsmoothed sources, namely $d \geq \Delta_s$. As long as the network load is such that d is less than Δ_s , the net saving is negative for the one-hop case. When d exceeds Δ_s , savings are non-negative for both one-hop and three-hops cases. Notice that the three-hops case starts obtaining non-negative savings at a smaller network utilization than one-hop case due to the extra savings at two more congested hops.

Figure 3.10 shows the effect of the smoothing interval (number of frames smoothed over) on savings in end-to-end delay $D - \hat{D}$ for the one-hop case at 18.8% and 33.2% network utilization levels. At 18.8% utilization level, smoothing is not effective at all, whereas for 33.2% utilization level, smoothing delays of up to 7 frame periods gives zero

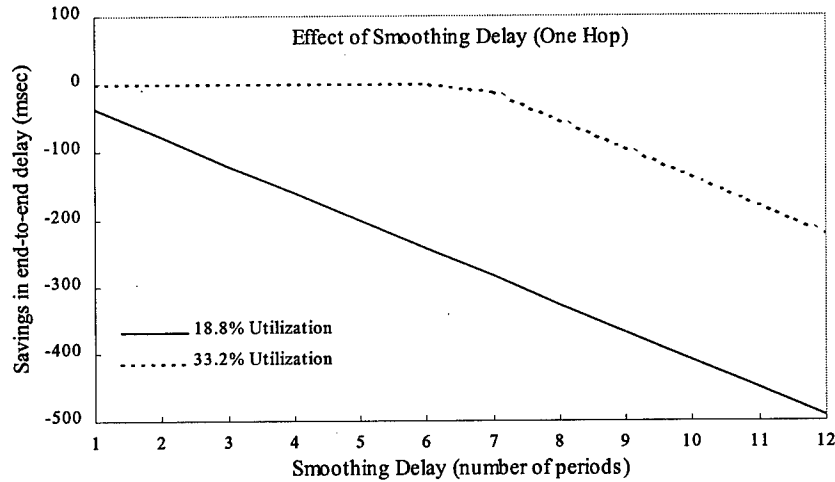


Figure 3.10: Effect of smoothing delay for one-hop network.

net gain while for longer smoothing delays, the queuing delay bound of 291.67 msec is exceeded for 33.2% utilization level. The results of the same experiments conducted for the three-hops case are shown in Figure 3.11. As in the case of one-hop, there is no positive saving at 18.8% utilization level. At 33.3% utilization level, positive savings are achieved and smoothing delay of 7 frame periods provides the maximum saving. As observed in Figure 3.8, savings increase with larger smoothing delays which explains the positive slope before the peak occurs. However, the saving starts to decrease for smoothing delays larger than 7 frame periods since the queuing delay is smaller than the smoothing delay. This indicates the importance of choosing the correct smoothing interval for a given network load according to the rules of Theorems 3.1 and 3.2 and Proposition 3.1 since improperly chosen smoothing policies can be far worse than doing no smoothing at all.

Finally, Figure 3.12 depicts the effect of smoothing delay on the total end-to-end delay bound. Network clients are interested in achievable end-to-end delay bound and it

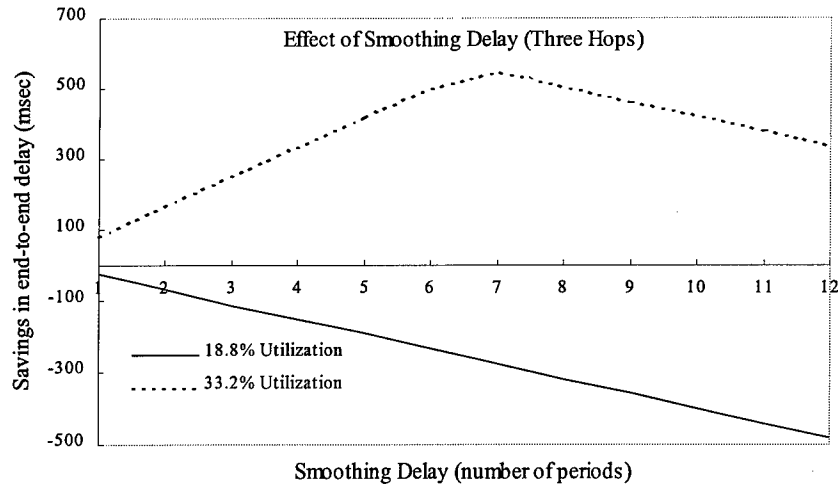


Figure 3.11: Effect of smoothing delay for three-hops network.

can be observed that choosing an improper smoothing interval can increase end-to-end delay. For example, if 300 msec end-to-end delay is required, the only admissible smoothing delays are 6, 7 and 8 frame periods at 33.2% network load.

3.5.2 Real-time Traffic

In order to evaluate the effect of smoothing on the network utilization for real-time

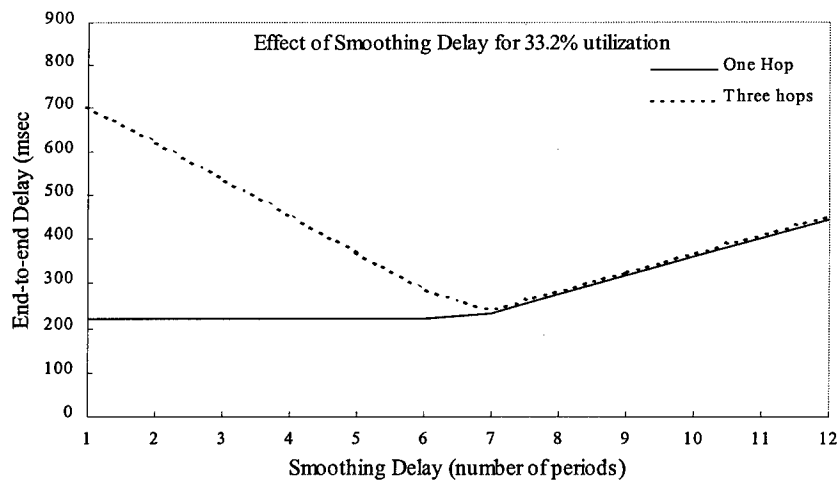


Figure 3.12: Effect of smoothing delay on total end-to-end delay bound.

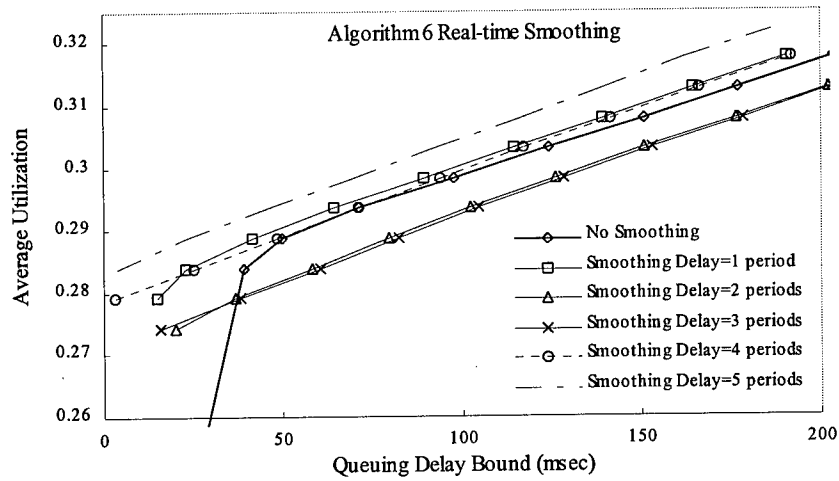


Figure 3.13: Utilization vs queuing delay for real-time smoothing.

traffic, the same set of experiments were conducted using the real-time smoothing algorithms presented in Chapter 2. Algorithms 6 and 7 correspond to the original and improved smoothing algorithms proposed for real-time traffic. Algorithm 7 is improved in terms of number of rate changes at the expense of higher peak rate and larger variation in the rate.

Figure 3.13 shows the network utilization as a function of queuing delay bound using Algorithm 6 for various smoothing delays. For smoothing delays of 2 and 3 frame periods, real-time smoothing does not decrease the queuing delay bound. This indicates that although real-time smoothing generates smoother traffic profile in general, there could be more arrivals within an interval due to the prediction error for future picture sizes, or equivalently $\hat{b}(t) > b(t)$ for some t . In Chapter 3, it has been shown that real-time smoothing algorithm performs the worst when smoothing delay is between 2 and 4 periods from the MBS point of view. However, for smoothing delay larger than 3 periods,

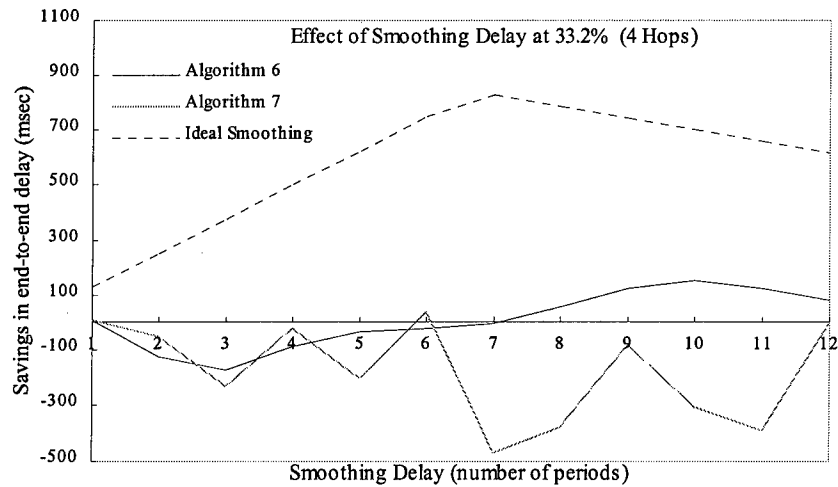


Figure 3.14: Smoothing delay vs. Savings in the end-to-end delay bound for real-time and ideal smoothing algorithms.

network utilization is increased indicating that real-time smoothing algorithm generates smoother traffic according to Definition 3.1.

In Figure 3.14, the effect of smoothing delay on the savings in end-to-end delay bound is depicted for real-time smoothing algorithms 6 and 7 at a network with four congested hops. The results indicate that from network utilization point of view, Algorithm 6 performs better than Algorithm 7, but can never match the performance of ideal

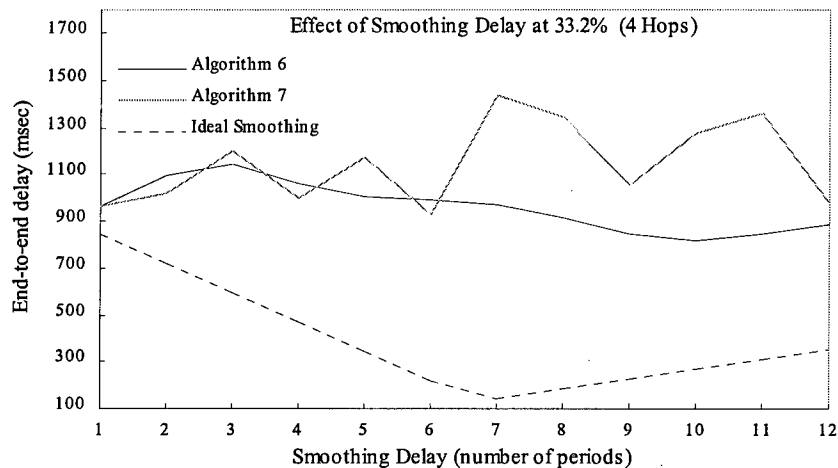


Figure 3.15: Effect of smoothing delay on the end-to-end delay.

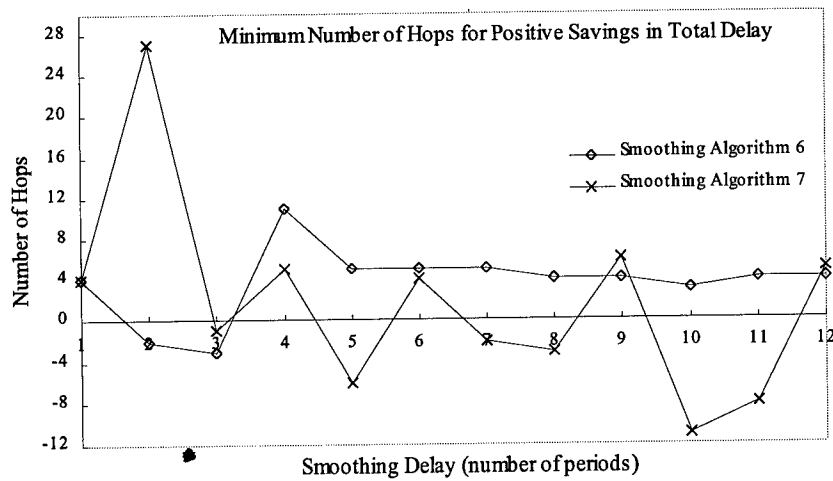


Figure 3.16: Minimum number of congested hops to obtain positive savings.

smoothing. Even for large smoothing delays, Algorithm 7 can not increase network utilization although it requires less demanding UPC than that of Algorithm 6. These results are in par with the observation made in Chapter 3 regarding the insufficiency of UPC parameters to express the network resource requirements of the source. When compared with Algorithm 6, Algorithm 7 costs more to the network in terms of allocated resources, but costs less to the client in terms of specified UPC. Figure 3.15 shows the effect of smoothing delay on the total end-to-end delay bound. It is clear that Algorithm 7 should not be used from network utilization point of view.

In Figure 3.16, the minimum number of congested hops to obtain positive savings in the end-to-end delay bound is presented as a function of smoothing delay. At negative values, network utilization is lower with respect to the case when unsmoothed sources are used. With the exception of smoothing delay of 3 periods, Algorithm 6 can always be used to decrease end-to-end delay bound if there exists a minimum number of congested hops in the path. For example, if the number of congested hops is at least 5, then Algo-

rithm 6 can be used with smoothing delays larger than 4 frame periods. However, with Algorithm 7, it is almost impossible to provide such a general rule since burstiness of the smoothed source is not bounded by the smoothing delay.

The results obtained in this chapter are valid for networks with FCFS scheduling discipline where packets are served according to the first-come first-served policy. Although other scheduling disciplines, such as the Round Robin (RR) or the Weighted Round Robin (WRR), could be as well considered, it was found in [79] that RR and WRR disciplines are ill suited for CBR traffic, both in terms of performance and implementation complexity. Therefore, FCFS scheduling discipline is believed to be a reasonable choice in the context of research pursued in this chapter since piecewise CBR traffic is generated by the smoothing function.

Finally, it should be noted that the reported results include the worst case measurements of queuing delay so network utilizations presented here are for deterministic service only. However, this does not prevent the utilization of unused network resources for services providing statistical or best-effort guarantees.

3.6 Conclusion

In this chapter, the effect of smoothing on the performance of networks with deterministic guarantees was investigated. It was shown both analytically and empirically that when ideal smoothing is used, positive savings in the end-to-end delay bound are obtained when there exists at least two congested hops in the path. Equivalently, the extra smoothing delay is equal to the gain in queuing delay when multiplexing homogenous sources at a congested hop. This is in contrast to the result found in a previous study

where smoothing results in no benefit to the network client for a single-hop case when traffic shaping is implemented by a FIFO with a constant service rate. However, when real-time traffic is submitted to the network, more congested hops are required in order to justify the benefit of smoothing. It was also found that current ATM UPC parameter set is not sufficient to express the actual requirements of the network clients from network cost point of view. It was illustrated that sources with less demanding UPC can actually cost more to the network in terms of allocated resources than less bursty sources.

From the network point of view, the benefit obtained by smoothing scheme corresponds to higher utilization of network capacity since more connections are admissible at a given QoS. This benefit may be realized with a better QoS (via a reduced delay bound) or a better price of service (via increasing the network utilization). The pricing scheme must encourage the clients to smooth their traffic for maximum benefit. This can be provided by incorporating price into the smoothing decision when a QoS is being requested by the client, which would presumably encourage some intended type of behavior.

Chapter 4

Aggregate Smoothing: Integration of Traffic Shaping and Multiplexing

4.1 Introduction

With the deployment of broadband integrated services based on the Asynchronous Mode Transfer (ATM) technology, it will be possible to provide a large variety of services and distributed applications. It is likely that video traffic will dominate due to the high bandwidth requirement of full motion HDTV-quality video transmission. To reduce the bandwidth needed, video is generally compressed by a video compression standard such as MPEG-1 and MPEG-2 [6, 8] before transmission. The compression method of a video stream can be either CBR compression where the output bit rate of the encoder is forced to be constant resulting in variable image quality, or VBR compression where the output bit rate varies according to the requirement of the underlying video sequence guaranteeing constant image quality. As described in Chapter 3, CBR transport of video is easier to manage from the network point of view since bandwidth allocation and tariff for network usage are simple. It is also straightforward for the network to multiplex several CBR channels onto a communication channel since the cells arrive at constant rate. How-

ever, it is difficult to statistically multiplex VBR video streams and guarantee lossless delivery of cells, since the bit rates of the multiplexed streams may peak together. Lossless cell delivery is not possible unless the peak bandwidth is allocated, in which case the delivery of the VBR stream will be expensive. For the case when several video streams are to be transported as a bundle over a channel, it is possible to shape or smooth each VBR stream such that the aggregate bandwidth of the channel is reduced. To distinguish this from *individual smoothing*, the term *aggregate smoothing* will be used. This chapter focuses on aggregate smoothing of a group of video streams to be delivered as a bundle given the individual constraints of each stream in terms of delay and buffer bounds.

Application areas of aggregate smoothing include video broadcasting, video-on-demand (VoD) and long-distance video-telephony service [11, 84]. In the case of video-broadcasting, all the video streams originating from a broadcasting center are to be delivered to all receivers. As illustrated in Figure 4.1(a), a single channel from the broadcasting center to the local fiber node can be used to deliver all the video streams which are then distributed to the households in the neighborhood. Figure 4.1(b) depicts the VoD scenario in which the video streams are not all destined to a common destination, but rather one video session is delivered to each household. This, however, does not preclude the use of video aggregation. In a public network, there is typically a distribution center to which many subscribers in a neighborhood are connected. The VoD server may be located in a central office and serve an area covered by several distribution centers. Video streams targeted to the same distribution node may be aggregated and at the distribution

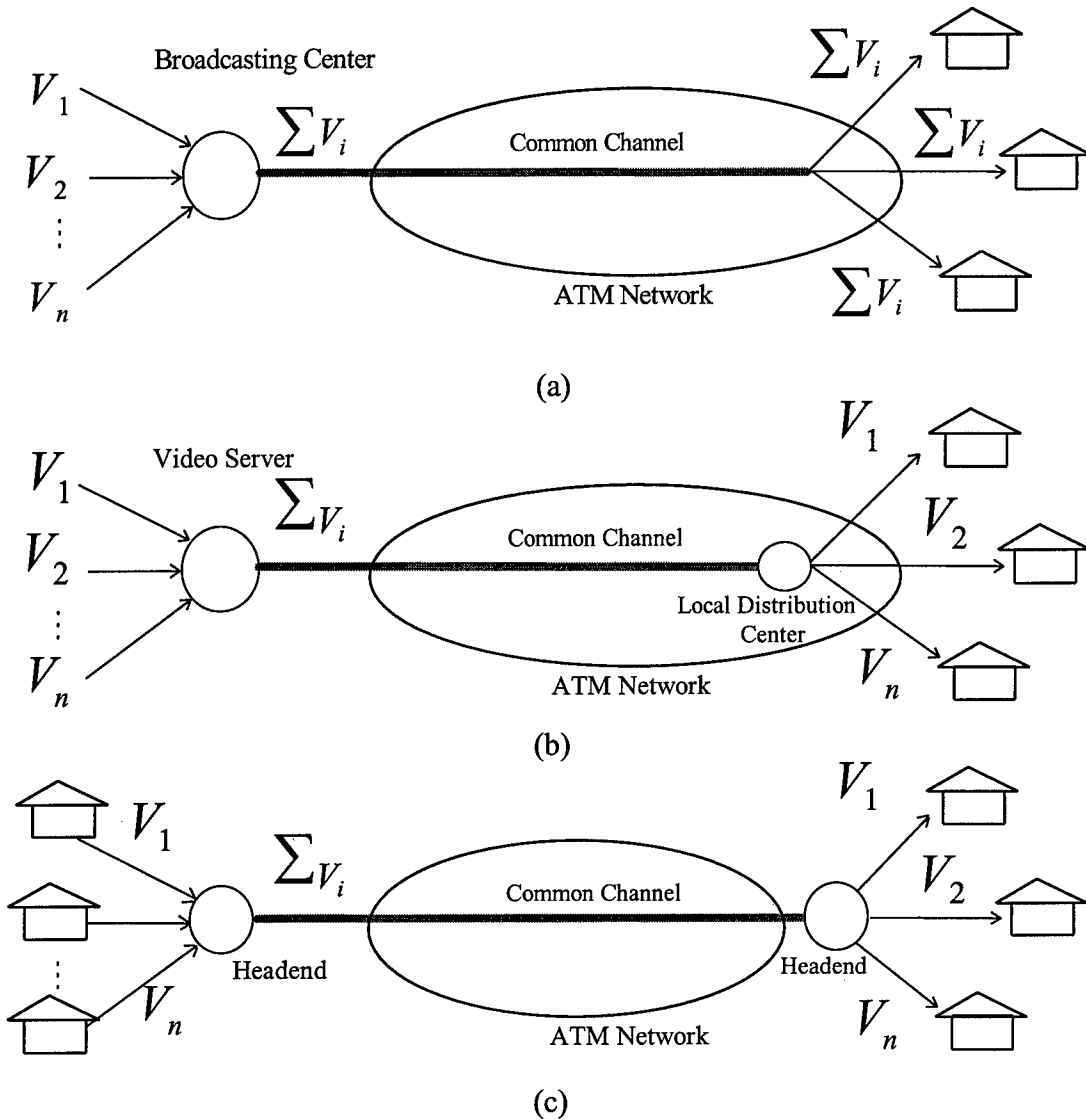


Figure 4.1: Applications of aggregate smoothing: (a) video broadcasting (b) Video-on-Demand (c) Video-telephony service.

node, they are separated and forwarded to their respective destinations. Figure 4.1(c) shows the long-distance video-telephony service scenario. This service provides point-to-point communications where sources and receivers are geographically separated. The video streams destined for a common remote area can be aggregated at the local network headend to where subscribers of the local region are connected. In this scenario, the goal of aggregation is to save expensive long-distance bandwidth.

In Chapter 2, smoothing algorithms that minimize the number of rate changes were developed so that renegotiation cost is reduced when CBR transport is used for VBR-compressed video. However, for a single video session with real-time traffic, it has also been found that renegotiation cost is high given the current processing capability of a typical ATM access switch. When several video streams are to be transported as a bundle, the number of rate changes for the common channel can be further reduced. Instead of smoothing each video session individually, the aggregate rate is smoothed. The resulting traffic profile corresponding to each video session is burstier than the case when individual smoothing is applied, however, when smoothed bit streams are multiplexed onto a common link, aggregate smoothing results in less number of rate changes and smoother traffic. Therefore, the bandwidth requirements and the number of rate changes are reduced for the combined traffic when aggregate smoothing is used. This scheme can be used by any video transport system with multiple video sources that can be multiplexed onto a common link. One particular feature of the aggregate smoothing is that, individual buffer and delay constraints are also guaranteed. To the author's knowledge, this is the first scheme that smoothes the aggregate rate and also satisfies the buffer and delays constraints of the multiplexed video streams.

From the network utilization viewpoint, the benefit of aggregate smoothing is obvious since the traffic profile corresponding to aggregate rate is smoother. However, from the individual stream viewpoint, the resulting traffic profile will be, in general, burstier compared to the case when individual smoothing is applied to each video stream as an independent process. This may affect the network QoS provided for that particular

stream. Therefore, the effect of aggregate smoothing on the network delay is investigated and it is found that significant savings in end-to-end delay bound can be achieved when network load is high.

Related work includes studies performing the statistical multiplexing and compression of several video streams in a related manner. In [80-82], a smoothing buffer is used to collect outputs from the video streams. The encoding parameters of the video streams are directly modified (e.g., the quantization scale) based on the CBR bandwidth constraint. The occupancy level of the buffer is used as the feedback to determine the amount of data the encoders may output in the future. The key idea is that when the buffer level is high, the encoders must encode at a lower image quality to prevent buffer overflow; and when the buffer level is low, the encoders can encode at a higher image quality. In contrast, constant quality (e.g., open-loop with no feedback from the network) encoding is assumed which allows for HDTV quality video transport without any loss. Also, one would not need to deal with the intricate problem of setting the appropriate feedback parameters during the system design in order to get reasonable image quality. References [83] and [84] do not use the smoothing-buffer feedback mechanism. In [83], the encoding parameters of the video streams are directly modified to reduce the bit rate, whereas in [84], data in each video stream is discarded according to some signal-to-noise or distortion metric in order to reduce the aggregate bit rate less than the reserved bit rate of the CBR channel. The scheme in [84] has the following disadvantages with respect to the aggregate smoothing. First, it is not suitable for no-loss, constant quality video communications. Second, it provides a mechanism for only MPEG compressed video streams.

Third, implementation complexity can be high for real-time processing since the unit of video aggregation is a slice period which is much smaller than a frame period. And finally, it is not clear how to determine CBR bandwidth required by a group of aggregated video streams such that cell loss rate can be bounded. Another work solves the minimum reservation rate problem for multiple pre-encoded MPEG video streams over a CBR channel [85]. However, their solution applies to VoD type of applications that simply playback stored MPEG video and require very large buffers at the receiver for a two-hour movie. Both in [84] and [85], a CBR channel is used for the whole session. On the other hand, RCBR scheme allows for better resource utilization and provides the true bandwidth as needed. With aggregate smoothing, the number of rate changes is significantly reduced such that RCBR transport can be used with little cost of rate renegotiation.

The rest of the chapter is organized as follows. In Section 4.2, the concept and description of aggregate smoothing is presented. In Section 4.3, the performance of aggregate smoothing is evaluated by using MPEG traces of stored and real-time video. In Section 4.4, the effect of aggregate smoothing on the network utilization is investigated and it is shown that aggregate smoothing allows the network to provide better QoS compared to the case when smoothing is applied to each video stream as an independent process. Finally, Section 4.5 concludes this chapter.

4.2 Specification of Aggregate Smoothing

In Figure 4.2, the system model for aggregate smoothing is shown. N video streams with constraints (D_i, B_i) are multiplexed onto a link where D_i denotes the extra

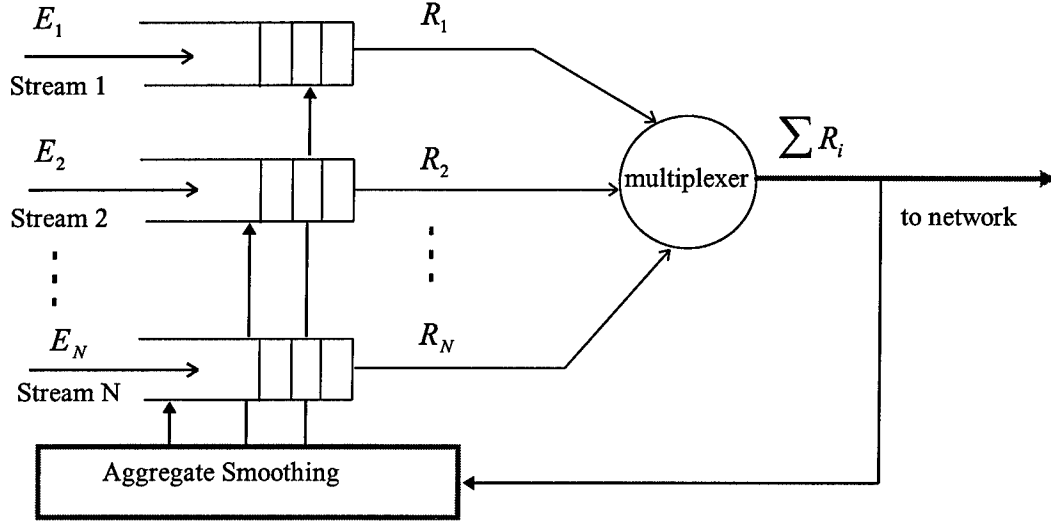


Figure 4.2: System model for aggregate smoothing.

smoothing delay bound in terms of frame period and B_i denotes the maximum buffer size in bits for stream i . Each stream has its own smoothing buffer and the size of frame that arrives to the buffer at the beginning of period k , belonging to stream i is denoted by E_k^i and its smoothed rate at the buffer output is denoted by R_k^i . A simple round-robin scheme is used at the multiplexer. The objective of aggregate smoothing is to smooth the multiplexer output given the set of constraints (D_i, B_i) for $i = 1, \dots, N$. The aggregate rate function at period k is defined as $R_k^A = \sum_{i=1}^N R_k^i$ and the rate vector as $\mathbf{R}_k^T = [R_k^1, R_k^2, \dots, R_k^N]_{N \times 1}$. In the following, aggregate smoothing algorithm is described.

Step 1:

Derive the upper and lower bounds for the cumulative aggregate rate function. Assume that minimum buffer size is specified as zero ($B_{\min,i}^e = 0$) and $B_{\max,i}^e = B_i$. The number of

frames at the buffer at time $t = 0$ is one. From Equation (2.7), the upper and lower bounds for bit stream i , $i = 1, \dots, N$, with constraints (D_i, B_i) can be written as:

$$L_j^i = \max \left(\sum_{k=1}^{j-D_i} E_k^i, \sum_{k=1}^{j+1} E_k^i - B_i \right) \text{ for } j > D_i \text{ and } U_j^i = \sum_{k=1}^j E_k^i \text{ for and } j \geq 1.$$

The cumulative of aggregate rate function can be then bounded by :

$$\sum_{i=1}^N L_j^i \leq \sum_{k=1}^j R_k^A \leq \sum_{i=1}^N U_j^i \quad (4.1)$$

Step 2:

Find the shortest path through the upper bounds given by $\sum_{i=1}^N U_j^i$ and the lower bounds given by $\sum_{i=1}^N L_j^i$ using the algorithm specified in Figure 2.4 to obtain R_k^A at the multiplexer output.

Step 3:

For $i = 1, \dots, N$, find the shortest path through L_j^i and U_j^i using the algorithm specified in Figure 2.4 to obtain the optimal smoothing rate \hat{R}_k^i for stream i . Let the optimal rate vector be defined as $\hat{\mathbf{R}}_k^T = [\hat{R}_k^1, \hat{R}_k^2, \dots, \hat{R}_k^N]_{N \times 1}$.

Step 4:

At the final step, derive \mathbf{R}_k from $\hat{\mathbf{R}}_k$ given the constraint that $\sum_{i=1}^N R_k^i = R_k^A$. The closest vector to the surface defined by $\sum_{i=1}^N R_k^i = R_k^A$ gives the best possible suboptimal rate vector in the Euclidean space. Then, R_k^i is given by r_{\min}^i where r_{\min}^i satis-

finds $\min_{\forall r^1, r^2, \dots, r^N} \sum_{i=1}^N (r^i - \hat{R}_k^i)^2$. To solve for r_{\min}^i , r^N is substituted with $R_k^A - \sum_{j=1}^{N-1} r^j$

and the minimum of $\sum_{i=1}^N (r^i - \hat{R}_k^i)^2$ is found by taking its derivative with respect to r^j

and solving for zero values where the minimum occurs. Then, for $i = 1, \dots, N-1$, R_k^i is given by

$$\begin{bmatrix} R_k^1 \\ R_k^2 \\ \vdots \\ R_k^{N-1} \end{bmatrix} = \frac{1}{\Delta_{N-1}} \cdot \begin{bmatrix} \Delta_{N-2} & -1 & \cdots & -1 \\ -1 & \Delta_{N-2} & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & \Delta_{N-2} \end{bmatrix} \begin{bmatrix} \hat{R}_k^1 + R_k^A - \hat{R}_k^N \\ \hat{R}_k^2 + R_k^A - \hat{R}_k^N \\ \vdots \\ \hat{R}_k^{N-1} + R_k^A - \hat{R}_k^N \end{bmatrix} \quad (4.2)$$

where Δ_j is computed recursively as $\Delta_j = 2 \cdot \Delta_{j-1} - (j-1)$ for $j > 2$ with the initial conditions $\Delta_0 = 1$ and $\Delta_1 = 2$. R_k^N is given by

$$R_k^N = R_k^A - \sum_{j=1}^{N-1} R_k^j. \quad (4.3)$$

However, R_k^i as derived in Equations (4.2) and (4.3) may be out of bounds for some i , that is $R_k^i > U_1^i$ or $R_k^i < L_1^i$ for some i . In Figure 4.3, two cases are illustrated. The first case is when the estimated aggregate rate R_k^A is larger than the sum of upper bound of R_k^i or smaller than the sum of lower bound of R_k^i as shown in Figure 4.3(a). This case occurs when prediction error for the future traffic is large resulting in overestimation or underestimation of aggregate bit rate. Therefore, the following conditions are checked before the closest Euclidean vector is derived in Step 4:

$$\text{if } R_k^A < \sum_{i=1}^N L_k^1, \text{ then choose } R_k^A = \sum_{i=1}^N L_k^1 \text{ and } R_k^i = L_k^1$$

or

if $R_k^A > \sum_{i=1}^N U_k^i$, then choose $R_k^A = \sum_{i=1}^N U_k^i$ and $R_k^i = U_k^i$.

If any of these conditions is true, Step 4 is skipped, otherwise, R_k^i is derived as in Step 4.

Figure 4.3(b) illustrates the second case where the rate vector found in Equations (4.2) and (4.3) is out of bounds for some i . In this case, the rate vector that satisfies the minimum boundary condition is found by traversing along the surface until the closer upper or lower boundary point is reached. This is done by calculating the net increase or decrease in the aggregate rate after out-of-bound rates are adjusted. In order for the aggregate rate not to change, other in-bound rates are also adjusted considering their respective upper and lower bounds.

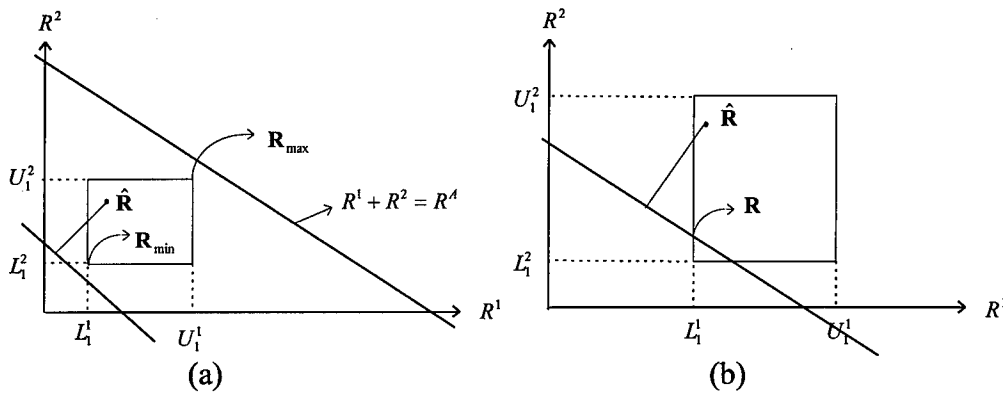
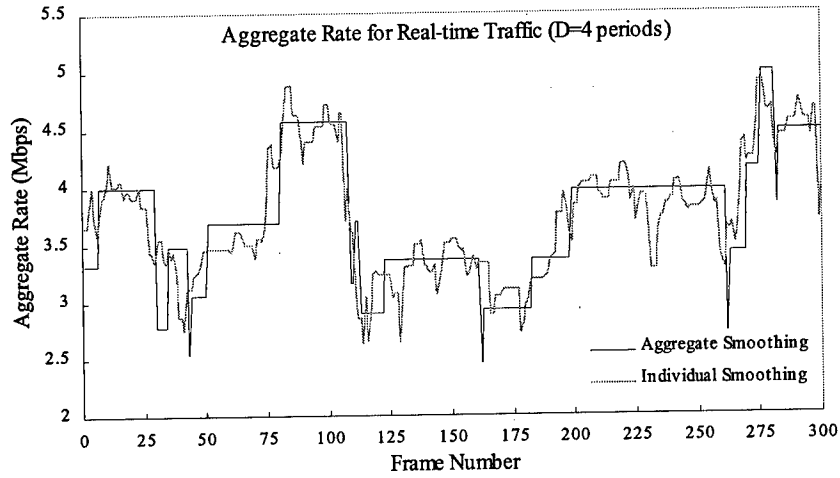
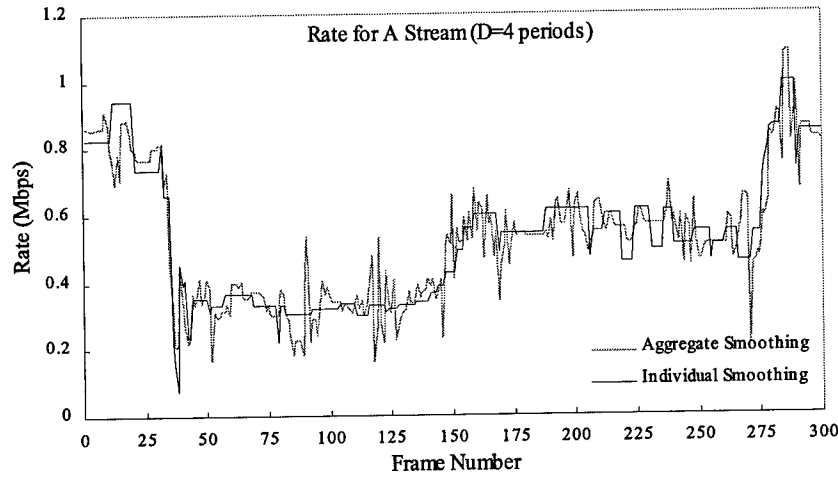


Figure 4.3: (a) Aggregate bit rate is out of bounds. (b) Closest rate vector is out of bounds.



(a)



(b)

Figure 4.4: (a) Rate function of total rate at the multiplexer output.
(b) Rate function of a single stream.

The complexity of aggregate smoothing is derived as follows. Steps 1 and 2 take $O(H)$ time where H is the size of the look-ahead window. Steps 3 and Step 4 take $N \cdot O(H)$ time (Step 4 takes constant time since it is a matrix multiplication operation). Then the total algorithm time is $(N + 1) \cdot O(H)$. In the case of individual smoothing, the total computation time is $N \cdot O(H)$. So the percentage increase in the algorithm complexity is $100 \times 1/N$. For large N , the increase in algorithm complexity is almost non-

significant. For example for $N = 20$ and 100, the complexity increases by only 5% and 1% respectively.

4.3 Evaluation of Aggregate Smoothing

Aggregate smoothing scheme is illustrated in Figure 4.4. Both aggregate and individual smoothing algorithms are applied to a group of 5 real-time video streams each of with smoothing delay of 4 frame periods. From Figure 4.4(a), it is observed that aggregate smoothing provides smoother rate function than individual smoothing for the total rate at the multiplexer output. However, as shown in Figure 4.5(b), the rate function of each stream is burstier for the case of aggregate smoothing since the rate determined by aggregate smoothing is suboptimal.

In order to compare the performance of aggregate smoothing to that of individual smoothing, MPEG traces of Formula 1 and Star Wars are used. Each trace consists of 36,000 frame sizes in bits corresponding to duration of 25-minutes at a display rate of 24 frames/sec. Three performance measures introduced in Chapter 2 are used which include the number of rate changes of aggregate rate, the peak aggregate rate and the standard deviation of aggregate rate.

In the experiments, homogenous sources are assumed where all video streams have different start times of the original stream. This is realized by shifting the original stream's arrival pattern by an amount $(i - 1) \cdot 36000 / N$ with the traces wrapped around to the beginning when they reach the end for $i = 1, \dots, N$. The smoothing delay is fixed to 4 periods with no buffer constraint for all video streams. The size of the look-ahead

window is 100 for stored video and 12 for real-time video. First, the effect of the number of video sessions on the aggregate bit rate is investigated. In Figure 4.5, the number of rate changes, peak rate and standard deviation of rate as a function of number of video sessions are presented for the case of stored video. It is observed that while aggregate smoothing keeps the number of rate changes almost constant, the number of rate changes increases for the case of individual smoothing. The benefit of aggregate smoothing can be realized even with only two video sessions. In general, when the number of multiplexed sources increases, the gain from statistical multiplexing also increases. However, this has little effect on the number of rate changes in the case of aggregate smoothing.

Aggregate smoothing results in higher peak rate than the case of individual smoothing due to the nature of algorithm which tends to keep constant rate as far as the constraints allow for. In fact, for a small number of video sessions less than 7, aggregate smoothing has less peak rate due to statistical multiplexing gain, however this gain is not so effective when the number of video sessions is over 7.

As expected, the standard deviation of total bit rate is less than the case when individual smoothing is applied. This is again related to aggregate smoothing's ability to utilize the statistical multiplexing gain which results in smoother rate at the multiplexer output. Therefore, better network utilization is obtained with aggregate smoothing.

The same set of experiments was conducted for the case of real-time video. Figure 4.6 shows the experimental results for the three performance measures. As in the case of stored video, aggregate smoothing reduces the number of rate changes dramatically, especially when the number of video sessions is large. For example, the number of rate

changes is decreased by a factor of 6 when 15 Star Wars video sessions are multiplexed. For a large number of video sessions, the aggregate rate function changes almost every period resulting in very high renegotiation cost from the network viewpoint. With aggregate smoothing, renegotiation cost is significantly reduced and it is almost constant independent from the number of video sessions. Thus, RCBR transport becomes a feasible solution when multiple video streams are sent as a bundle. However, from the network utilization viewpoint, aggregate smoothing does not guarantee better performance since the standard deviation of aggregate rate function is higher for some cases. In Figure 4.6, aggregate smoothing is outperformed by individual smoothing for Formula 1 trace, but it performs almost the same for Star Wars trace. Both smoothing schemes are not advantageous over another from the peak rate viewpoint. Based on the experimental results, it can be concluded that aggregate smoothing for real-time traffic is beneficial from RCBR transport viewpoint, although network utilization is lower than the case of individual smoothing. However, if the issue is to choose a transport service with deterministic guarantees, then RCBR transport service is a better choice since it provides the highest network utilization when compared to other services, e.g., regular CBR or VBR transport. Therefore, the use of aggregate smoothing is justified from the network utilization point of view for real-time traffic.

The effect of smoothing delay on the performance of aggregate smoothing is investigated by conducting a second set of experiments using stored and real-time video. In these experiments, the number of video sessions per multiplexer is fixed to 5. The results are similar for both stored and real-time video so only the experimental results for stored

video are presented in Figure 4.7. As expected, the relative performance of aggregate smoothing over individual smoothing decreases with larger smoothing delays. An interesting observation is that the number of rate changes seems to be independent from the statistics of video sources when smoothing delay is larger than 4 periods as both Formula 1 and Star Wars video sequences have almost the same number of rate changes. This allows for approximate estimation of renegotiation cost before the start of transmission given the number of video sessions.

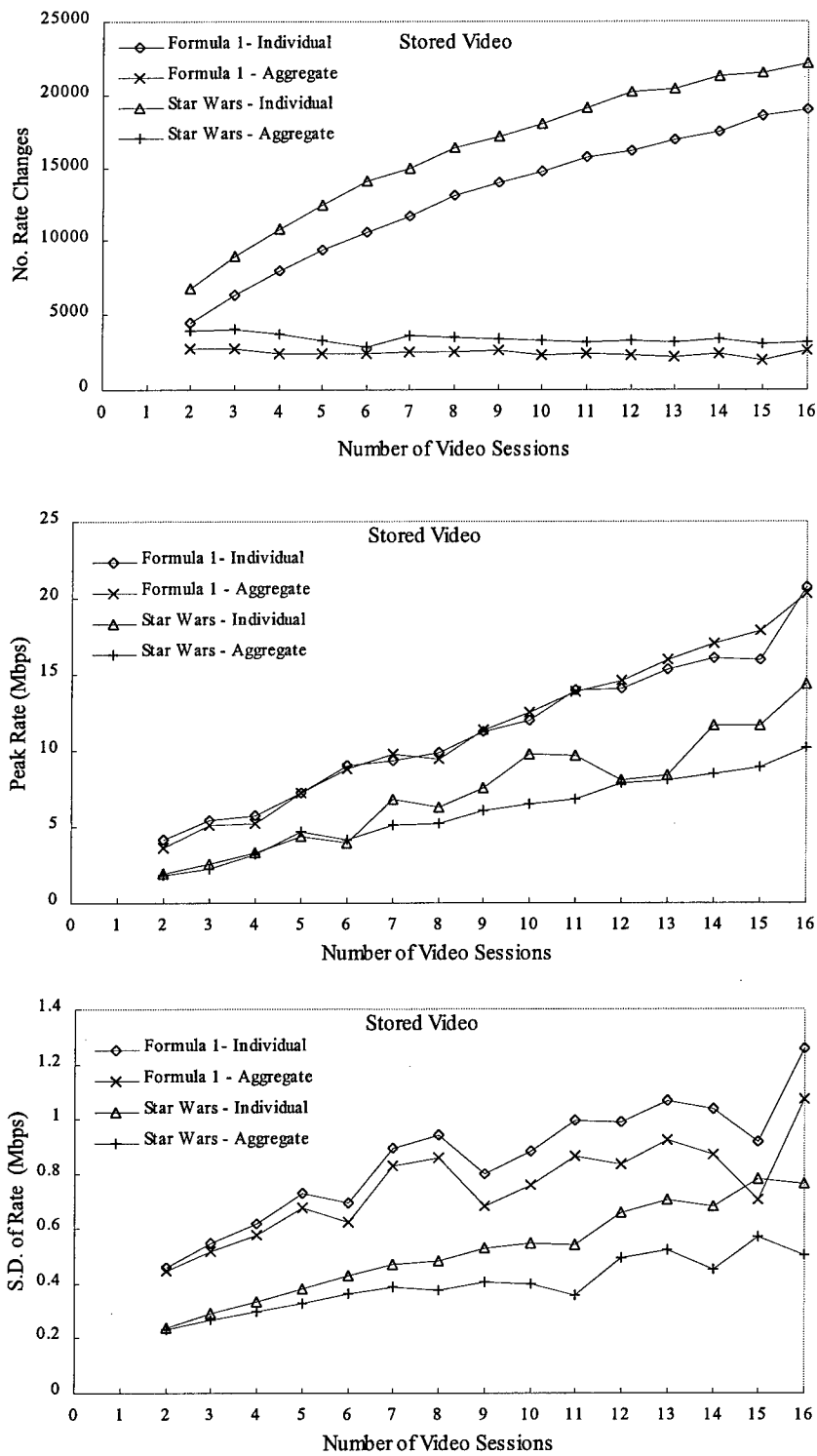


Figure 4.5: Performance of the aggregate smoothing algorithm for stored video.

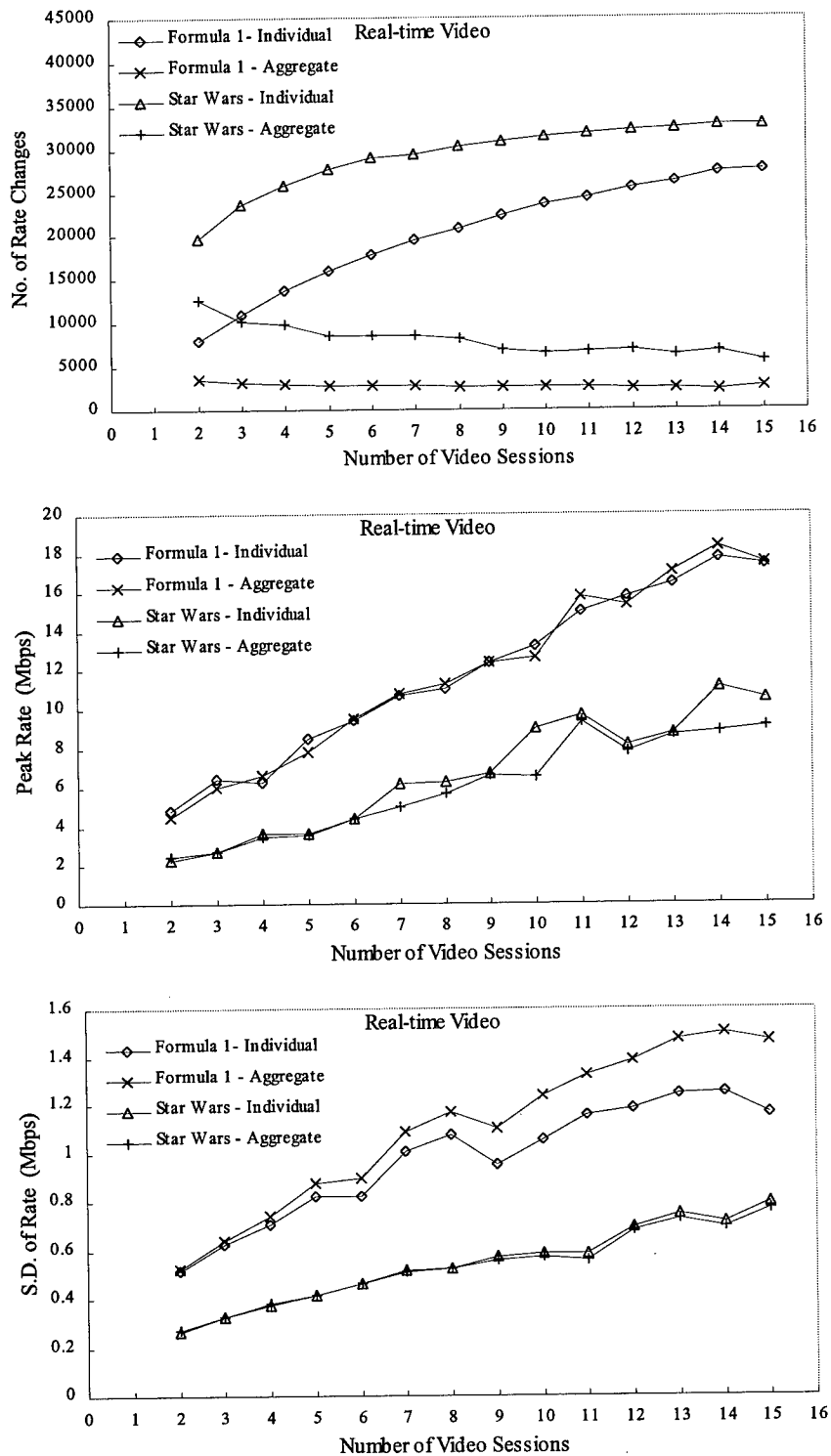


Figure 4.6: Performance of the aggregate smoothing algorithm for real-time video.

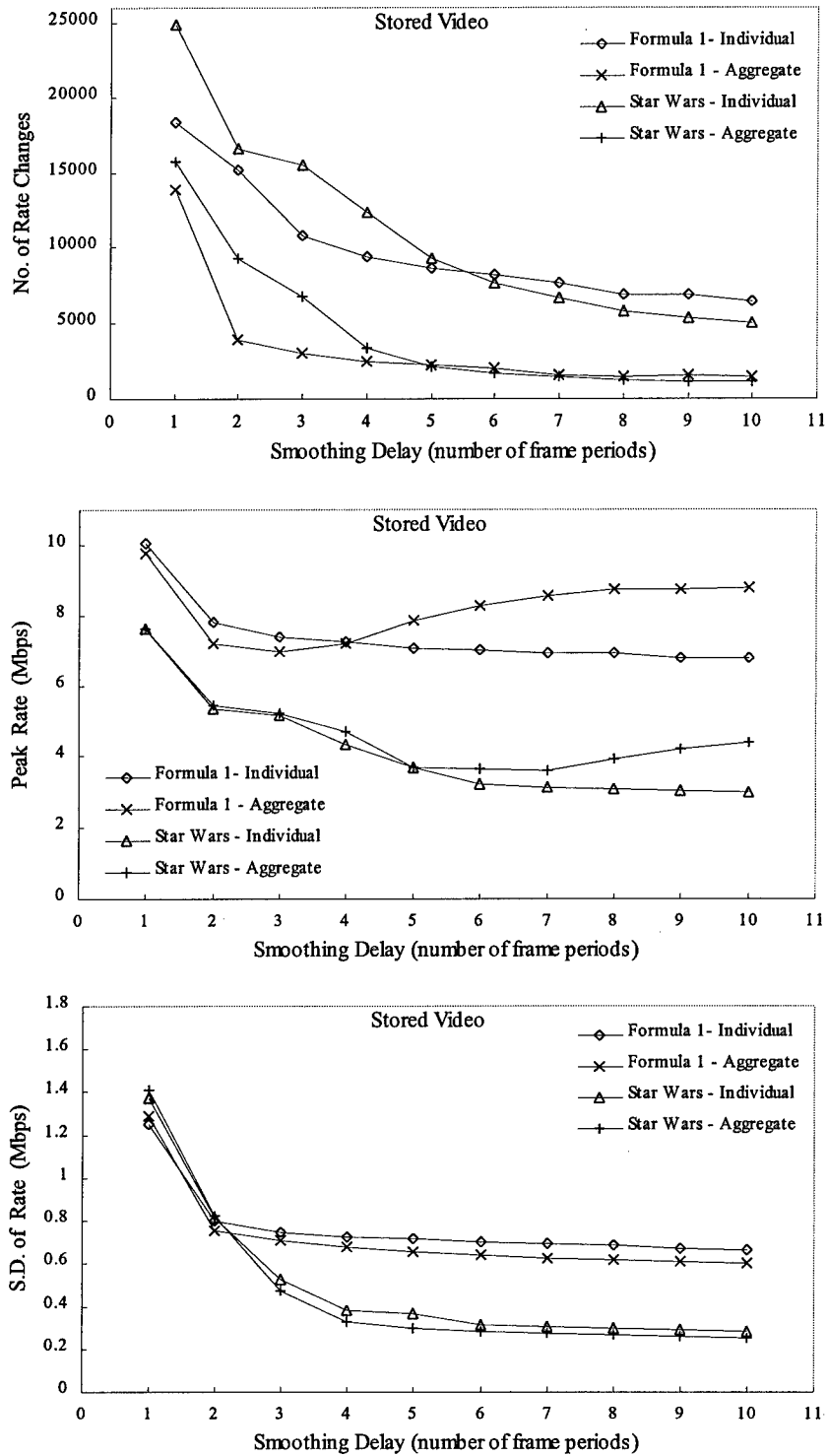


Figure 4.7: Effect of smoothing delay on the performance of aggregate smoothing.

4.4 Effect of Aggregate Smoothing on End-to-end Deterministic Guarantees

In Section 4.3, aggregate smoothing has been shown to decrease the number of rate changes and standard deviation of aggregate rate at the multiplexer output compared to the case when smoothing is applied to each video stream as an independent process. Assume that all channels in the network carry a group of multiplexed video streams. From the network utilization viewpoint, aggregate smoothing is beneficial compared to the case of individual smoothing since the aggregate rate function at each multiplexer output is smoother resulting in higher network utilization as described in Chapter 4. However, from the individual stream viewpoint, the resulting traffic profile for that particular stream will be burstier. It is measured that the peak rate of a stream in the case of aggregate smoothing can be 60% more than the case when individual smoothing is applied to that stream. It is also observed that the standard deviation increases by more than 8%. This affects the network QoS as perceived by a particular stream.

MPEG traces of Formula 1 and Star Wars video streams each with 10,000 frames are used in the experiments in order to investigate effect of aggregate smoothing on end-to-end deterministic delay bound. The video streams are first multiplexed onto a link in a round-robin fashion with individual or aggregate smoothing applied at the multiplexer. The packets of each group of streams are then served according to the FCFS service discipline at a link speed of 620 Mbps. Figure 4.8 illustrates the simulation scenario. The traces at each round-robin multiplexer have different start times as described in Section 4.3, and wrap around to the beginning when they reach the end of the original trace.

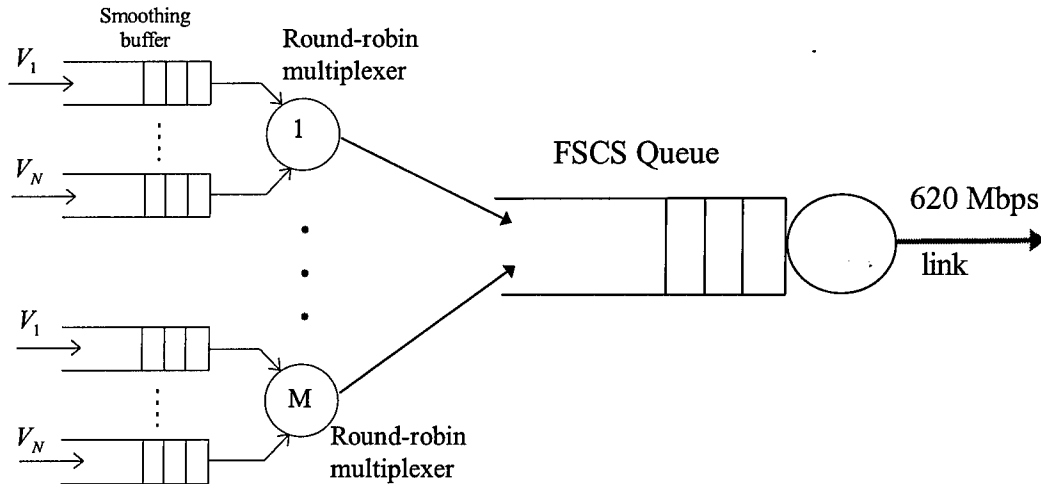
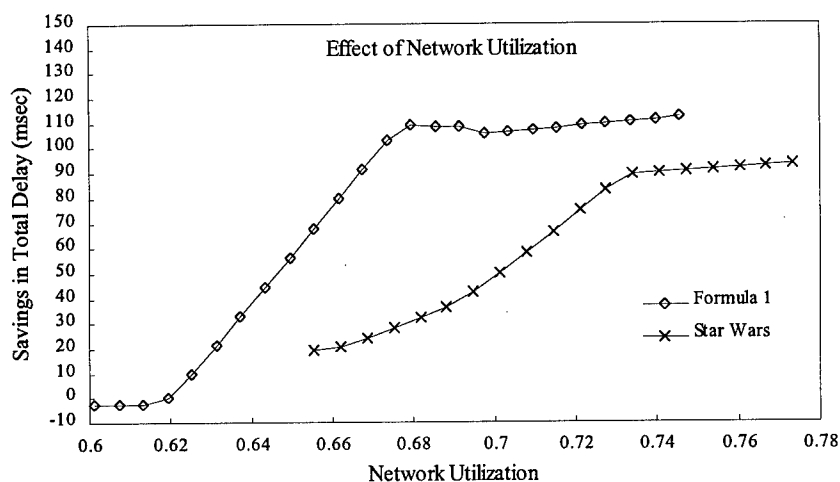


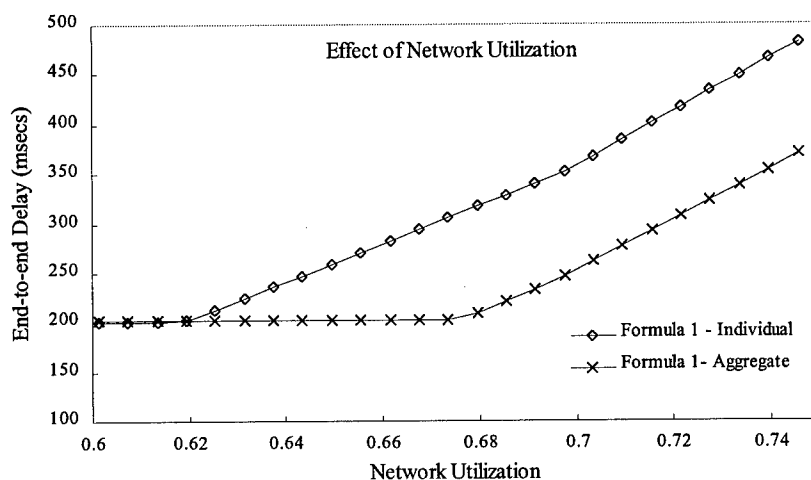
Figure 4.8: Scenario for trace-driven simulation of video aggregation scheme.

Therefore, only homogenous sources are considered both at the multiplexer and switch level. The purpose of these experiments is to compare the performance of the aggregate smoothing to that of individual smoothing from the network QoS viewpoint.

In the first set of experiments, the effect of network utilization on end-to-end delay bound is investigated for stored video. The number of streams per multiplexer is 5 and the smoothing delay is fixed as 4 frame periods for all streams. Figure 4.9 shows the experimental results for both Formula 1 and Star Wars traces. When the network is not congested, individual smoothing performs better due to the smoother traffic it generates for each stream. However, when the network load is increased, aggregate smoothing results in smaller delay bound such that significant savings in end-to-end delay bound can be achieved for a single hop. For example, in Figure 4.9(a), at 68% network utilization, savings in end-to-end delay is 110 msec for the Formula 1 trace. In Figure 4.9(b), end-to-end delay bound as a function of network utilization is shown. Aggregate smoothing provides 200 msec of end-to-end delay bound when the network load is 68%, whereas



(a)



(b)

Figure 4.9: Effect of network utilization on (a) the savings in end-to-end delay bound (b) the end-to-end delay.

individual smoothing can provide the same bound only when the network load is 62%.

This corresponds to almost 10% increase in the network utilization.

In the second set of experiments, the effect of smoothing delay on end-to-end delay bound is investigated for a given network utilization. Figure 4.10 shows the experimental results for both Formula 1 and Star Wars traces. The results indicate that savings in end-to-end delay bound depend on the smoothing delay and there is an optimal value of

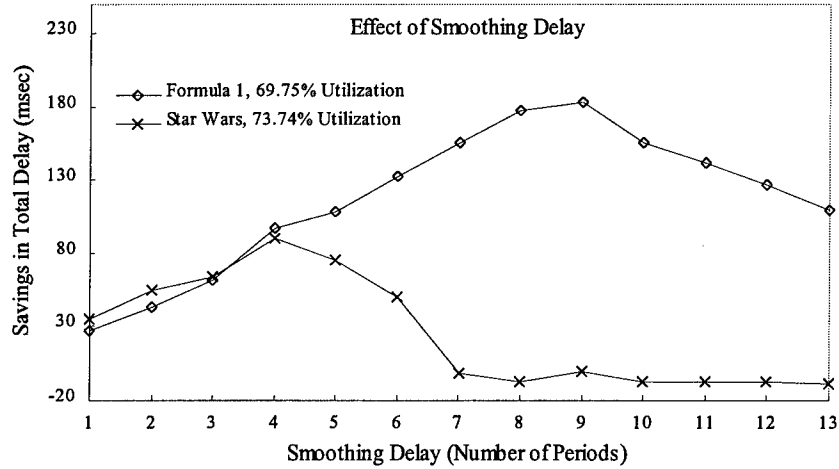


Figure 4.10: Effect of smoothing delay on the savings in end-to-end delay.

smoothing delay for which maximum savings can be obtained. For the case of Formula 1 trace at 69.75% network utilization, smoothing delay of 9 periods gives the maximum savings whereas for Star Wars trace at 73.74% network utilization, smoothing delay of 4 periods should be used. For larger values of smoothing delay, savings can be negative indicating that individual smoothing outperforms aggregate smoothing. However, even for that case, amount of increase in end-to-end delay is small enough to justify the use of aggregate smoothing for all network utilization levels.

4.5 Conclusion

This chapter has introduced and described a new concept called aggregate smoothing that integrates multiplexing and smoothing of multiple video sources grouped together and transmitted as a bundle. The novel feature of aggregate smoothing is its ability to satisfy individual requirements of each video stream (in terms delay and buffer constraints) while smoothing multiplexer output. It has been shown experimentally that aggregate smoothing can reduce the number of rate changes significantly such that RCBR

transport for real-time traffic can be efficient for multiple video sessions grouped together. From the network viewpoint, aggregate smoothing provides smoother traffic, thus higher network utilization compared to the case when individual smoothing is applied to each video stream as an independent process. These benefits are provided at little extra computational cost that is non-significant when the number of video sessions is over 20.

Chapter 5

Conclusion and Future Work

The burstiness of variable bit rate (VBR) traffic makes it difficult to efficiently utilize network resources, as well as to provide guaranteed end-to-end network quality of service (QoS) to the traffic sources. Smoothing or shaping the traffic at the entrance of the network reduces the burstiness thus allowing for higher utilization within the network since less network resource is required for the smoothed traffic. In this report, a methodology has been proposed that provides an efficient algorithm for smoothing of live or stored VBR traffic given a set of delay and buffer constraints. The efficiency of the proposed smoothing algorithm has been demonstrated by integrating it with a bandwidth allocation scheme that allows for better characterization of network resource requirements which in turn results in higher network utilization and lower transmission cost. The rest of the chapter is organized as follows. Section 5.1 summarizes the work presented in this report. Section 5.2 lists the main contributions of the dissertation. Section 5.3 concludes the chapter with future work where a number of issues are identified which can be pursued in the future using the framework developed in this report.

5.1 Overview of Presented Work

In the first chapter, an overview of communications requirements of multimedia applications was given along with a comparison of ATM network transport services that can be used for VBR traffic. Among the possible services, RCBR service requires the least number of rate changes in order to reduce renegotiation cost, whereas VBR transport service requires the smoothest traffic (with little variation in the rate) for efficient use of network resources to achieve the maximum statistical multiplexing gain when several video sources are multiplexed onto the same transmission link. Therefore, a smoothing algorithm that can utilize the underlying network services in a most efficient way and also can address the diverse requirements of network clients should be provided. The rest of the report describes the design and specification of such a smoothing algorithm and its effect on the network utilization.

In the second chapter, a smoothing algorithm for lossless transmission of VBR traffic was introduced for both real-time and stored data. A novel feature of the algorithm is its ability to provide a unique solution to diverse requirements of applications expressed in terms of delay and buffer bounds. It has been shown that the algorithm is effective in smoothing real-time VBR traffic when its performance is compared with respect to other techniques existing in the literature because of a novel approach that minimizes the effect of future traffic prediction error on the rate function by decoupling past and future information from each other as far as possible. This important contribution has been demonstrated by conducting a large number of experiments using MPEG compressed video sequences. Even with a rudimentary forecasting rule, the causal algorithm was shown to

be as effective as the ideal smoothing scheme where future traffic is known. The proposed scheme allows to minimize either the number of rate changes or the standard variation of the rate function depending on what network transport service is available, e.g., RCBR or VBR service.

In Chapter 3, the effect of smoothing on deterministic end-to-end delay bound was investigated. For the case of ideal smoothing where future traffic is known, it is shown both analytically and empirically that the extra delay contributed by smoothing is equal to the saving in queuing delay when multiplexing homogenous sources at a congested hop. This indicates that, with ideal smoothing, it is possible to achieve higher network utilization without any degradation in the QoS of the connection even for a single hop. Alternatively, for multiple congested nodes, smoothing results in significant reductions in the end-to-end delay bound since sum of the savings in queuing delay at each congested hop is more than the incurred extra smoothing delay at the source. This result is particularly important since it proves that there exists a smoothing scheme with no negative effect on the network utilization. It was also demonstrated that sources with less demanding UPC can actually cost more to the network in terms of allocated resources than less bursty sources.

Finally, the fourth chapter introduces and describes a new concept called *aggregate smoothing* that integrates smoothing and multiplexing of multiple video sources grouped together and transmitted as a bundle. The novel feature of aggregate smoothing is that individual constraints of each video session (delay and buffer bounds) are satisfied while smoothing the total rate at little extra computational cost that is non-significant when the

number of video sessions is over 20. It has been shown experimentally that aggregate smoothing reduces the number of rate changes significantly such that RCBR transport for real-time traffic can be a cost-effective network service for multiple video sessions grouped together. From the network viewpoint, aggregate smoothing provides smoother total traffic, thus, higher network utilization can be achieved compared to the case when individual smoothing is applied to each video stream as an independent process.

5.2 Contributions

The main contributions of this report are summarized below:

- 1) A lossless smoothing algorithm was introduced and specified. The unique features of the algorithm design that distinguish it from other approaches are:
 - The algorithm is not optimized for a specific set of constraints. Instead, the specification of constraints is independent from the algorithm design which allows for a wide set of constraints be specified and imposed by the multimedia applications.
 - The algorithm performance can be optimized for a specific performance measure which allows for better use of underlying network services.
 - The effect of traffic estimation error on the algorithm performance was minimized by a novel approach which computes the upper bounds using estimates of future traffic and the lower bounds using the history. Therefore, accurate estimation of future traffic is not as much re-

quired to achieve good performance when compared with other approaches.

- 2) It was proved both analytically and empirically that ideal smoothing allows the network to support more connections with the same end-to-end QoS guarantees for any number of hops in packet-switching networks.
- 3) A new concept called *aggregate smoothing* was introduced which integrates multiplexing of VBR sources with smoothing. This is the first technique that solves the problem of smoothing aggregated traffic and satisfying the individual constraints of each source. It was shown that aggregate smoothing makes it possible to utilize RCBR service for real-time traffic and at the same increases the network utilization.

5.3 Future Work

In this section, some research issues that may be pursued using the framework developed in this report are identified as in the following.

- (1) In the experiments, MPEG video has been used to represent VBR traffic due to its acceptance as the standard for digital compressed video and since it is expected that future broadcasting and network services will use MPEG encoded bit stream for video transmission. However, as stated in Chapter 2, the proposed smoothing algorithm can be used with arbitrary traffic that provides an estimation rule for the future. For MPEG video, a simple estimate that uses the size of picture in previous GOP, gives good results since pictures j and $j - N$ are of the same type (I, B or P). An estimation rule must be

determined for other traffic by either observation of statistical behavior of traffic or utilizing the traffic structure as in the case of MPEG video. Simple estimation rule for MPEG video will not be sufficient for applications supporting full VCR functionality through functions such as backwards, fast-forward or scanning due to the different type of traffic generated by each mode. Therefore, an estimation rule must be determined and passed to smoothing function so that future traffic can be predicted for best performance.

(2) The proposed smoothing algorithm can be used as part of a video transport system where the QoS of the underlying network services may change during the transmission. For networks with no QoS guarantees, the algorithm can be extended to include varying network conditions as well. Since the foundation of the model is based on the status of buffers at the server and client, the algorithm can adapt to changing network conditions by using the feedback from the client to recompute the bounds based on information about the buffer occupancies at the client and server.

(3) In Chapter 3, only networks with deterministic guarantees on end-to-end performance are considered when investigating the effect of smoothing on the network utilization. The study could be extended to include networks that provide statistical guarantees in addition to deterministic guarantees on the QoS of the connection. The effect of smoothing on network utilization can be better understood with a more realistic network modeling. Another area where the results of Chapter 4 can be used is in the design of networks that would encourage their clients to smooth their traffic in order to achieve maximum utilization of network resources. Therefore, new connection admission control policies and pricing schemes must be developed for better network resource management.

BIBLIOGRAPHY

- [1] S. E. Minzer, "Broadband ISDN and Asynchronous Transfer Mode (ATM)," *IEEE Communications Magazine*, pp. 17-24, September 1989.
- [2] H. Armbruster and K. Wimmer, "Broadband Multimedia Applications Using ATM Networks: High-performance Computing, High-Capacity Storage, and High-speed Communication," *IEEE Journal on Selected Areas in Communications*, Vol. 10, No. 9, December 1992.
- [3] D. Hehmann, M. Salmony, and H. J. Stuttgen, "Transport Services for Multimedia Applications on Broadband Networks," *Computer Communications Review*, Vol. 13, No.4, pp. 197-203, May 1990.
- [4] H. Leopold, "Classification of Multimedia Types," *OSI95 Report*, OSI95/ELIN/D1/03/TN/R/V1, Alcatel, May 1991.
- [5] T. Köprülü, K. Cardakli, and C.Y. R. Chen, "Multimedia Communication Requirements," To appear in the *Journal of Parallel and Distributed Computing Systems*.
- [6] D. LeGall, "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of the ACM*, Vol. 34, No. 4, pp. 47-58, April 1991.
- [7] M. L. Liou, "Overview of the $p \times 64$ Kbps Video Coding Standard," *Communications of the ACM*, Vol. 34, No.4, pp. 60-63, April 1991.
- [8] MPEG-2 Systems Committee, "MPEG-2 Systems Working Draft," *ISO/IETC/JTC1/SC29/WG-11-N0501*, Seoul, July 1993.

- [9] H. Stuttgen, "Network Evolution and Multimedia Communication," *IEEE Multimedia*, Vol. 2, No. 3, pp. 42-59, Fall 1995.
- [10] W. Tawbi, F. Horn, E. Horlait and J. B. Stefani, "Video Compression Standards and Quality of Service," *The Computer Journal*, Vol. 36, No. 1, pp. 43-54, January 1993.
- [11] T. Kwok, "A Vision for Residential Broadband Services: ATM-to-the-Home," *IEEE Network*, Vol. 9, No. 5, pp. 14-28, September 1995.
- [12] O. Rose. "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," *Institute of Computer Science Research Report Series Report No. 101*, University of Wuerzburg., February 1995.
- [13] S. S. Lam, S. Chow, and D. K.Y. Yau, "A Lossless Smoothing Algorithm for Compressed Video," *IEEE/ACM Transactions on Networking*, Vol.4, No.5, pp. 697-708, October 1996.
- [14] M. de Prycker, R. Peschi, and T. VanLangedem, "B-ISDN and the OSI Protocol Reference Model," *IEEE Network*, Vol. 7, No. 2, pp. 10-18, March 1993.
- [15] The ATM Forum, *ATM: User-Network Interface Specification*. Eaglewood Cliffs, NJ: Prentice-Hall, 1993.
- [16] J. M. McManus and K. W. Ross, "Video-on-Demand over ATM: Constant-Rate Transmission and Transport," *IEEE Journal in Selected Areas in Communications*, Vol. 14, No. 6, pp. 1087-1098, August 1996.
- [17] J. Lauderdale, "Variable Bit-Rate Video Transmission over ATM Networks," M.Phil Thesis, Chapter 3 "The Minimum Reservation Rate Problem for the Transmission of MPEG VBR Video," The Hong Kong University of Science and Technology, Dept. of Electrical and Electronic Engineering. August 1995.

- [18] M.W. Garret and Walter Willinger, "Analysis, Modeling and Generation of Self-Similar VBR Video Traffic," In *Proceedings of ACM SIGCOMM '94*, pp. 269-280, University College London, London, UK, August 1994.
- [19] E. P. Rathgeb, "Policing of Realistic VBR Video Traffic in an ATM Network," *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, pp. 325-334, April 1991.
- [20] M. Grossglauser, S. Keshav, and D. Tse, "RCBR: A Simple and Efficient Service for Multiple Time-Scale Traffic," In *Proceedings of ACM SIGCOMM '95*, pp. 219-230, Cambridge, Massachusetts, USA, August 1995.
- [21] H. Harasaki and M. Yano, "A Study on VBR Coder Control under Usage Parameter Control," In *Proc. 5th Int. Packet Video Workshop*, Berlin, Germany, 1993.
- [22] J. Y. Hui, "Resource Allocation for Broadband Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 9, December 1988.
- [23] J. S. Turner, "Managing Bandwidth in ATM Networks with Bursty Traffic," *IEEE Network*, September 1992.
- [24] B. Doshi and S. Dravida, "Congestion Controls for Bursty Data Traffic in Wide Area High Speed Networks: In-Call Negotiations," In *Proceedings of ITC Specialist Seminar 7*, Morristown, NJ, 1990.
- [25] S. Chong, S.Q. Li, and J. Ghosh, "Predictive Dynamic Bandwidth Allocation for Efficient Transport of Real-Time VBR Video over ATM," *IEEE Journal on Selected Areas in Communications*, Vol. 13, No.1, pp. 12-23, January 1995.
- [26] H. Zhang and E.W. Knightly, "A New Approach to Support Delay-Sensitive VBR Video in Packet-Switched Networks," In *Proceedings of 5th Workshop on Network-*

- ing and Operating System Support for Digital Audio and Video*, pp. 275-286, April 1995.
- [27] D. J. Reininger, D. Raychaudhuri, and J. Y. Hui, "Bandwidth Renegotiation for VBR Video Over ATM Networks," In *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 6, pp. 1076-1085, August 1996.
- [28] P.E. Boyer and D.P. Tranchier, "A Reservation Principle with Applications to the ATM Traffic," *Computer Networks and ISDN Systems*, Vol.24, pp. 321-334, 1992.
- [29] T. Köprülü, K. Cardakli, D. Meliksetian, and C.Y. R. Chen, "A New Approach to Bandwidth Renegotiation for VBR Video: Integration of Traffic Shaping and Bandwidth Allocation," *Technical Report MNS-96-903*, Syracuse University, September 1996.
- [30] K. Sriram, "Methodologies for Bandwidth Allocation, Transmission Scheduling, and Congestion Avoidance in Broadband Networks," *Computer Networks and ISDN Systems*, Vol. 26, No. 1, pp. 43-59, 1993.
- [31] K. Nahrstedt and J.M. Smith, "Revision of QoS Guarantees at the Application/Network Interface," *Technical Report*, University of Pennsylvania, Distributed Systems Laboratory, PA 19104-6389, 1994.
- [32] L. Delgrossi, C. Halstrick, D. Hehmann, R.G. Herrtwich, O. Krone, J. Sandvoss, and C. Vogt, "Media scaling for audio-visual communication with the Heidelberg transport system," in *Proceedings of ACM Multimedia '93*, pp. 99-104, August 1993.

- [33] H. Kanakia, P. Mishra, and A. Reibman, "An adaptive congestion control scheme for real-time packet video transport," in *Proceedings of ACM SIGCOMM '93*, pp. 20-31, September 1993.
- [34] P. Pancha and M. El Zarki, "Bandwidth requirements for variable-bit rate MPEG sources in ATM networks," in *Proceedings of INFOCOM '93*, pp. 902-909, March 1993.
- [35] T. Ott, T. Lakshman, and A. Tabatai, "A scheme for smoothing delay-sensitive traffic offered to ATM networks," in *Proceedings of INFOCOM '92*, pp. 776-785, 1992.
- [36] W.C. Feng and S. Sechrest, "Critical bandwidth allocation for the delivery of compressed video," *Computer Communications*, Vol. 18, No. 10, pp. 709-717, October 1995.
- [37] A. R. Reibman and B.G. Haskell, "Constraints on variable bit-rate video for ATM networks," *IEEE Trans. Circuits and Syst. for Video Tech.*, Vol. 2, No. 4, pp. 361-372, December 1992.
- [38] D. T. Lee and F. P. Preparata, "Euclidean shortest paths in the presence of rectilinear barriers," *Networks*, Vol 14, pp. 393-410, 1984.
- [39] T. Köprülü, K. Cardakli, D. Meliksetian, and C.Y.R. Chen, "A lossless smoothing algorithm for compressed VBR video," *Technical Report MNS-96-801*, Syracuse University, August 1996.
- [40] G.C. Goodwin and K.S. Sin, *Adaptive Filtering Prediction and Control*, Prentice Hall, 1984.

- [41] I. Hsu and J. Walrand, "Quick Detection of Changes in Traffic Statistics: Application to Variable Rate Compression," in *Proceedings of the 32nd Allerton Conference on Communications, Control and Computing*, Monticello, IL, 1993.
- [42] CCITT Recommendation MPEG-1, "Coded Representation of Picture, Audio, and Multimedia/Hypermedia Information," *ISO/IEC 11172*, Geneva Switzerland, 1993.
- [43] E. W. Knightly, and P. Rossaro, "Effects of Smoothing on End-to-end Performance Guarantees for VBR Video," In *Proceedings of the 1995 International Symposium on Multimedia Communications and Video Coding*, New York, NY, October 1995.
- [44] E.W. Knightly, and P. Rossaro, "Smoothing and Multiplexing Tradeoffs for Deterministic Performance Guarantees to VBR Video," *Technical Report TR-95-033*, International Computer Science Institute, Berkeley, CA, July 1995.
- [45] L. Georgiadis, R. Guerin, and V. Peris, "The effect of Traffic Shaping in Efficiently Providing End-to-end Performance Guarantees," *Technical Report RC 20014*, IBM Research Division, Yorktown Heights, NY April 1995.
- [46] N. Yamanaka, Y. Sato, and K. Sato, "Traffic Shaping for VBR Traffic in ATM Networks," *IEICE Transactions Communications*, E75-B(10):1105-1108, October 1992.
- [47] G. De Veciana and J. Walrand, "Traffic Shaping for ATM Networks," *Technical Report UCB/ERL M92/135*, University of California at Berkeley, Berkeley, CA, December 1992.
- [48] J. Lauderdale, D.H.K. Tsang, and G. Baciuc, "An Improved Algorithm for Lossless Smoothing of MPEG Video with Delay Constraints," *Visual 96 International Conference on Visual Information Systems*, Melbourne, Australia, February 1996.

- [49] K. Joseph and D. Reininger, "Source Traffic Smoothing and ATM Network Interfaces for VBR MPEG Video Encoders," In *Proceedings of 1995 IEEE International Conference on Communications*, Vol. 3, pp. 1761-1767, June 1995.
- [50] D. Reininger, G. Ramamurthy, and D. Raychaudhuri, "VBR MPEG Video Coding with Dynamic Bandwidth Renegotiation," In *Proceedings of 1995 IEEE International Conference on Communications*, Vol. 3, pp. 1773-1778, June 1995.
- [51] ATM Forum, Technical Working Group, "SAA Audio-Visual Multimedia Service (AMS) Implementation Agreement," *Contribution 95-0012*, Burlingame, CA, February 1995.
- [52] W. Verbiest, L. Pinnoo, and B. Voeten, "The impact of the ATM concept on Video Coding," *IEEE Journal in Selected Areas in Communications*, Vol. 6, No. 9, pp. 1623-1632, 1988.
- [53] D. G. Morrison, "Variable Bit-Rate Video Coding for Asynchronous Transfer Mode Networks," *Br. Telecom Technol. J.*, Vol. 8, No 3, July 1990.
- [54] D. Reininger and D. Raychadhuri, "Bit-rate Characteristics of a VBR MPEG Encoder for ATM Networks," In *Proceedings of IEEE ICC '93*, Geneva, Switzerland, May 1993.
- [55] D. Reininger, "Statistical Multiplexing of VBR MPEG Compressed Video on ATM Network," In *Proceedings of IEEE INFOCOM '93*, San Fransisco, CA, March 1993.
- [56] F. Kishino, "Variable Bit-Rate Coding of Video Signals for ATM Networks," *IEEE Journal in Selected Areas in Communications*, Vol. 7, No. 5, pp. 801-806, June 1989.

- [57] H. Sun *et al.*, "Error Concealment Algorithms for Robust Decoding of MPEG Compressed Video," to appear in *IEEE Transactions in Image Processing*.
- [58] S. El-Henaoui and Samir Tohme, "UPC Parameters for Real-Time VBR MPEG Traffic Applying Dynamic Bandwidth Allocation," In *Proceedings of IEEE ICC '96*, Dallas, June 1996.
- [59] J. Lauderdale and D. H.K. Tsang, "Bandwidth Scheduling of Prerecorded VBR Video for ATM Networks," Presented at *IEEE ATM Workshop*, Washington D.C., October 1995.
- [60] MPEG data. Trace available via anonymous FTP from [thumper.bellcore.com](ftp://thumper.bellcore.com).
- [61] M. W. Garrett and A. Fernandez, *Variable Bit Rate Video Bandwidth Trace Using MPEG Code*, November 1994. Available via anonymous FTP from [thumper.bellcore.com](ftp://thumper.bellcore.com).
- [62] G.M. Ramamurthy, *Private Communication*, C&C Research Labs, NEC USA, July 1994.
- [63] A. Eleftheriadis and D. Anastassiou, "Meeting Arbitrary QoS Constraints Using Dynamic Rate Shaping of Coded Digital Video," *Proc. 5th Intl. Workshop on Network and Operating System Support for Digital Audio and Video*, Durham, New Hampshire, April 18-21, 1995, pp. 95-106.
- [64] H. Sun, W. Kwok, and J. Zdepski, "Architectures for MPEG Compressed Bitstream Scaling," in *Proc. IEEE Int. Conf. Image Processing*, Washington, D.C., October 1995.

- [65] D.G. Morrison, M.E. Nilsson, and M. Ghanbari, "Reduction of the Bit-rate of Compressed Video while in its coded Form," In *Proc. Sixth Int. Workshop Packet Video*, Portland, September 1994.
- [66] P. Pancha and M. El Zarki, "Leaky Bucket Access Control for VBR MPEG Video Control," In *Proceedings of IEEE Infocom 95*, Boston, MA, April 1995.
- [67] H. Zhang and E. Knightly, "RED-VBR: A Renegotiation-Based Approach to Support Delay-Sensitive Video," To appear in *ACM/Springer-Verlag Multimedia Systems Journal*.
- [68] I. Richardson and M. Riley, "Usage Parameter Control Cell Loss Effects on MPEG video," In *Proceedings of IEEE ICC '95*, pp. 970-974, Seattle, WA, June 1995.
- [69] E. Knightly and H. Zhang, "Traffic Characterization and Switch Utilization Using Deterministic Bounding Interval Dependent Traffic Models," In *Proceedings of IEEE INFOCOM '95*, pp. 1137-1145. Boston, MA, April 1995.
- [70] C. Chang, "Stability, Queue Length, and Delay of Deterministic and Stochastic Queueing Networks," *IEEE Transactions on Automatic Control*, Vol. 39, No. 5, pp. 913-931, May 1994.
- [71] R. Cruz, "A Calculus for Network Delay, Part I: Network Elements in Isolation," *IEEE Transactions in Information Theory*, Vol. 37, No. 1, pp. 114-121, January 1991.
- [72] E. Knightly, D. Wrege, J. Liebeherr, and H. Zhang, "Fundamental Limits and Tradeoffs for Providing Deterministic Guarantees to VBR Video Traffic," In *Proceedings of ACM SIGMETRICS '95*, Ottawa, Ontario, May 1995.

- [73] H. Zhang and D. Ferrari, "Improving Utilization for Deterministic Service in Multimedia Communication," In *Proceedings of 1994 International Conference on Multimedia Computing and Systems*, pp. 295-304, Boston, MA, May 1994.
- [74] S. Low and P. Varaiya, "Burstiness Bounds for Some Burst Reducing Servers," In *Proceedings of IEEE INFOCOM '93*, pp. 2-9, San Fransisco, CA, March 1993.
- [75] N. Shroff and M. Schwartz, "Video Modeling within Networks Using Deterministic Smoothing at the Source," In *Proceedings of IEEE INFOCOM '94*, pp. 342-349, Toronto, Ontario, June 1994.
- [76] P. Skelly, M. Schwartz, and S. Dixit, "A Histogram-based Model for Video Traffic Behavior in an ATM Multiplexer," *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, pp. 446-459, August 1993.
- [77] D. Ferrari and D. Verma, "A Scheme for Real-time Channel Establishment in Wide-area Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 3, pp. 368-379, April 1990.
- [78] H. Zhang and D. Ferrari, "Rate-controlled Service Disciplines," *Journal of High Speed Networks*, Vol. 3, No. 4, pp. 389-412.
- [79] M. Grossglauser and S. Keshav, "On CBR Service," In *Proceedings of IEEE INFOCOM '96*, pp. 129-137, San Fransisco, CA, March 1996.
- [80] T. Koga, Y. Ijima, and T. Ishiguro, "Statistical Performance Analysis of an Inter-frame Encoder for Broadcast Television Signals," *IEEE Transactions in Communications*, Vol. 29, No.12, pp. 1868-1876, December 1981.

- [81] B.G. Haskell and A. Reibman, "Multiplexing of Variable Rate Encoded Streams," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 4, No.4, pp. 417-424, August 1994.
- [82] A. Guha and D. J. Reininger, "Multichannel Joint Rate Control of VBR Encoded Video for DBS Applications," *IEEE Transactions in Consumer Electronics*, Vol. 40, No. 3, August 1994.
- [83] G. Keesman and D. Elias, "Analysis of Joint Bit-Rate Control in Multi-Program Image Coding," In *Proceedings of SPIE Visual Communications Image Processing*, pp. 1906-1917, 1994.
- [84] S. C. Liew and C. Tse, "Video Aggregation: Adapting Video Traffic for Transport Over Broadband Networks by Integrating Data Compression and Statistical Multiplexing," *IEEE Journal on Selected Areas in Communications*, Vol. 14, No.6, pp.1123-1137, August 1996.
- [85] J. Lauderdale and D. H. K. Tsang, "The Minimum Reservation Rate Problem for the Transmission of MPEG VBR Video," Submitted to *ACM Sigmetric 96*.
<http://www.ee.ust.hk/~ustatm/index.html>

MISSION OF ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

- a. Conducts vigorous research, development and test programs in all applicable technologies;
- b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;
- c. Provides a full range of technical support to Air Force Material Command product centers and other Air Force organizations;
- d. Promotes transfer of technology to the private sector;
- e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.